IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of :

Yukie GOTOH et al. :

Serial No. NEW : **Attn: APPLICATION BRANCH**

Filed January 23, 2004 : Attorney Docket No. 2004-0089A

COMMON KEY EXCHANGING METHOD
AND COMMUNICATION DEVICE

CLAIM OF PRIORITY UNDER 35 USC 119

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Applicants in the above-entitled application hereby claim the date of priority under the International Convention of Japanese Patent Application No. 2003-015866, filed January 24, 2003, as acknowledged in the Declaration of this application.

A certified copy of said Japanese Patent Application is submitted herewith.

Respectfully submitted,

Yukie GOTOH et al.

By _____
Nils E. Pedersen
Registration No. 33,145
Attorney for Applicants

NEP/krg
Washington, D.C. 20006-1021
Telephone (202) 721-8200
Facsimile (202) 721-8250
January 23, 2004

THE COMMISSIONER IS AUTHORIZED TO CHARGE ANY DEFICIENCY IN THE FEES FOR THIS PAPER TO DEPOSIT ACCOUNT NO. 23-0975

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: ２００３年　１月２４日

出願番号
Application Number: 特願２００３－０１５８６６

[ST. 10/C]: [ＪＰ２００３－０１５８６６]

出願人
Applicant(s): 松下電器産業株式会社

２００３年１１月２８日

特許庁長官
Commissioner,
Japan Patent Office 今井康夫

出証番号　出証特２００３－３０９８６６４

【書類名】　特許願
【整理番号】　2032740136
【提出日】　平成15年 1月24日
【あて先】　特許庁長官殿
【国際特許分類】　H04L　9/00
【発明者】
【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内
【氏名】　五島　雪絵
【発明者】
【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内
【氏名】　横田　博史
【発明者】
【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内
【氏名】　玉井　昌朗
【発明者】
【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内
【氏名】　辻　敦宏
【発明者】
【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内
【氏名】　高垣　景一
【特許出願人】
【識別番号】　000005821
【氏名又は名称】　松下電器産業株式会社

【代理人】

【識別番号】　100097445

【弁理士】

【氏名又は名称】　岩橋　文雄

【選任した代理人】

【識別番号】　100103355

【弁理士】

【氏名又は名称】　坂口　智康

【選任した代理人】

【識別番号】　100109667

【弁理士】

【氏名又は名称】　内藤　浩樹

【手数料の表示】

【予納台帳番号】　011305

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【包括委任状番号】　9809938

【書類名】　明細書

【発明の名称】　共有鍵交換方法および通信機器

【特許請求の範囲】

【請求項１】　暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、

前記２つの通信機器は、第１の通信機器と第２の通信機器から成り、

前記第１の通信機器および前記第２の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、

前記第１の通信機器は、自分が生成した第１の秘密情報を基に第１の公開値を計算し、前記第２の通信機器に前記第１の公開値を送信し、

前記第２の通信機器は、第２の公開値を送信するまでの時間が遅くなることを応答予告として、前記第１の公開値の受信前または受信直後に、前記第１の通信機器に予め通知しておき、その後、自分が生成した第２の秘密情報を基に前記第２の公開値を計算し、前記第１の通信機器に前記第２の公開値を送信し、

前記第１の通信機器は、前記第２の通信機器から前記応答予告を受け取るようにしたことを特徴とする共有鍵交換方法。

【請求項２】　暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、

前記２つの通信機器は、第１の通信機器と第２の通信機器から成り、

前記第１の通信機器および前記第２の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、

前記第１の通信機器は、自分が生成した第１の秘密情報を基に第１の公開値を計算し、前記第２の通信機器に前記第１の公開値を送信し、

前記第２の通信機器は、第２の公開値を送信するまでの遅延時間または送信予定時刻を応答予告として、前記第１の公開値の受信前または受信直後に、前記第１の通信機器に予め通知しておき、その後、自分が生成した第２の秘密情報を基に前記第２の公開値を計算し、前記第１の通信機器に前記第２の公開値を送信し

、

前記第１の通信機器は、前記第２の通信機器から前記応答予告を受け取ると、前記第２の通信機器から今後送られてくるはずの前記第２の公開値を、少なくとも前記応答予告に基づく時間以上待機することを特徴とする共有鍵交換方法。

【請求項３】　前記第１の通信機器は、共有鍵交換の手順に先だって、前記第２の通信機器に前記応答予告を問い合わせるメッセージを発行し、前記第２の通信機器は前記メッセージに対して、第２の公開値を送信するまでの遅延時間または送信予定時刻を応答予告として、前記第１の通信機器に送信することを特徴とする、請求項１または請求項２に記載の共有鍵交換方法。

【請求項４】　前記第２の通信機器は、第２の公開値を算出するための処理時間を事前に実測しておき、前記第１の通信機器に前記応答予告を通知する際には、記録しておいた実測の処理時間を基に前記応答予告である遅延時間または送信予定時刻を算出することを特徴とする、請求項２または請求項３に記載の共有鍵交換方法。

【請求項５】　暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、

前記２つの通信機器は、第１の通信機器と第２の通信機器から成り、

前記第１の通信機器および前記第２の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、

前記第１の通信機器は、自分が生成した第１の秘密情報を基に第１の公開値を計算し、前記第２の通信機器に前記第１の公開値を送信し、

前記第１の通信機器は、タイムアウト時間より小さい再送間隔時間毎に、第２の通信機器に対して状況確認を行ない、

前記第２の通信機器は、後で送信する旨の通知を、前記第１の通信機器に行ない、

前記第１の通信機器は、後で送信する旨の通知を受信した場合、タイムアウト時間をリセットし、新たにタイムアウト時間の計測を始めるようにしたことを特徴とする共有鍵交換方法。

【請求項６】　暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、

前記２つの通信機器は、第１の通信機器と第２の通信機器から成り、

前記第１の通信機器および前記第２の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、

前記第２の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を前記第１の通信機器に通知し、

前記第１の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間と、通知された前記第２の通信機器が公開値の生成と共有鍵の生成に必要な時間とから、共有鍵交換処理を開始すべき時刻を算定し、

前記第１の通信機器は、前記共有鍵交換処理を開始すべき時刻までに、共有鍵交換処理を開始することを特徴とする共有鍵交換方法。

【請求項７】　暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、

前記２つの通信機器は、第１の通信機器と第２の通信機器から成り、

前記第１の通信機器および前記第２の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、

前記第２の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算出し、

前記第２の通信機器は、前記必要な時間から、共有鍵交換処理を開始すべき時刻を算定し、

前記第２の通信機器は、前記共有鍵交換処理を開始すべき時刻までに、前記第１の通信機器に、共有鍵交換処理の開始を要求することを特徴とする共有鍵交換方法。

【請求項８】　前記共有鍵交換処理を開始すべき時刻は、共有鍵の生成を完了すべき時刻よりも、前記公開値の生成と共有鍵の生成のために自身に必要な時間の２倍以上前であることを特徴とする請求項７に記載の共有鍵交換方法。

【請求項９】　暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、

前記２つの通信機器は、第１の通信機器と第２の通信機器から成り、

前記第１の通信機器および前記第２の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、

前記第１の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算出し、前記必要な時間から、自身が共有鍵交換処理を開始すべき時刻を算定し、

前記第２の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算出し、前記必要な時間から、自身が共有鍵交換処理を開始すべき時刻を算定し、

前記第１の通信機器と第２の通信機器のうち、前記共有鍵交換処理を開始すべき時刻が早い方が、遅い方に共有鍵交換処理の開始を通知し、

前記第１の通信機器と第２の通信機器は、共有鍵交換処理を開始することを特徴とする共有鍵交換方法。

【請求項１０】　暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、

前記２つの通信機器は、第１の通信機器と第２の通信機器から成り、

前記第１の通信機器および前記第２の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、

前記第１の通信機器は、自分が生成した第１の秘密情報を基に第１の公開値を計算し、前記第２の通信機器に前記第１の公開値を送信し、

前記第２の通信機器は、前記第１の通信機器から送られた第１の公開値が、既に存在する鍵を更新するための要求か否かを判定し、

鍵を更新するための要求と判定した場合には、自分が生成した第２の秘密情報を基に前記第２の公開値を計算する処理手順を小単位に分割し、さらに小単位の処理を時間的に負荷分散させて実行し、前記第１の通信機器に前記第２の公開値

を送信することを特徴とする共有鍵交換方法。

【請求項１１】　前記第２の通信機器は、第２の公開値を時間的に負荷分散させて算出する際の処理時間を事前に算出して、前記第１の通信機器に通知しておき、

前記第１の通信機器は、前記第２の通信機器から通知された前記処理時間を基に、鍵の更新のために共有鍵交換の処理を開始するタイミングを算出し、前記タイミングの時点までに共有鍵交換を開始することを特徴とする、請求項１０に記載の共有鍵交換方法。

【請求項１２】　暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、

前記２つの通信機器は、第１の通信機器と第２の通信機器から成り、

前記第１の通信機器および前記第２の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、

前記第２の通信機器は、次回の共有鍵交換処理が、既に存在する鍵を更新するための処理か否かを判定し、

鍵を更新するための処理と判定した場合には、自身が第２の秘密情報を基に前記第２の公開値を計算する処理手順と共有鍵を生成する処理手順とを、小単位に分割し、さらに小単位の処理を時間的に負荷分散させて実行し、前記第１の通信機器に前記第２の公開値を送信することを特徴とする共有鍵交換方法。

【請求項１３】　暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、

前記２つの通信機器は、第１の通信機器と第２の通信機器から成り、

前記第１の通信機器および前記第２の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、

前記第１、または、第２の通信機器は、共有鍵の演算を終了後、演算終了の通知を、相手の通信機器に行ない、相手の通信機器は、演算終了の通知を受けるまで、タイムアウト処理を行なわないようにすることを特徴とする共有鍵交換方法

。

【請求項１４】　暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、

前記２つの通信機器は、第１の通信機器と第２の通信機器から成り、

前記第１の通信機器および前記第２の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、

前記第２の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算定し、

前記第１の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算定し、

上記、必要な時間の少なくとも一方の値を基に、共有鍵の寿命の値を決定することを特徴とする共有鍵交換方法。

【請求項１５】　暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

自身が生成した第１の秘密情報を基に第１の公開値を計算し、第２の通信機器に第１の公開値を送信し、

第２の通信機器が行なう、第２の公開値を計算し送信するまでの時間が遅くなることを応答予告とする通知を、受信することを特徴とする第１の通信機器。

【請求項１６】　暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

自身が生成した第１の秘密情報を基に第１の公開値を計算し、第２の通信機器に第１の公開値を送信し、

第２の通信機器が、第２の公開値を送信するまでの遅延時間または送信予定時刻を応答予告として、第１の公開値の受信前または受信直後に、予め行なう通知

を受信しておき、その後、第２の通信機器自身が生成した第２の秘密情報を基に計算した第２の公開値を受信し

第２の通信機器から前記応答予告を受け取ると、第２の通信機器から今後送られてくるはずの第２の公開値を、少なくとも前記応答予告に基づく時間以上、受信できる状態で待機することを特徴とする第１の通信機器。

【請求項１７】　前記第１の通信機器は、共有鍵交換の手順に先だって、前記第２の通信機器に前記応答予告時間を問い合わせるメッセージを発行し、前記第２の通信機器が応答メッセージとして送信する第２の公開値を送信するまでの遅延時間または送信予定時刻を、前記応答予告として受信することを特徴とする、請求項１５または請求項１６に記載の第１の通信機器。

【請求項１８】　前記第２の通信機器に、第２の公開値を算出するための処理時間を事前に実測させておき、前記第１の通信機器に前記応答予告を通知する際には、記録しておいた実測の処理時間を基に前記応答予告である遅延時間または送信予定時刻を算出させることを特徴とする、請求項１６または請求項１７に記載の第１または第２の通信機器。

【請求項１９】　暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

自身が生成した第１の秘密情報を基に第１の公開値を計算し、第２の通信機器に第１の公開値を送信し、

タイムアウト時間より小さい再送間隔時間毎に、第２の通信機器に対して状況確認を行ない、

第２の通信機器が送信する、後で送信する旨の通知を、受信し、

後で送信する旨の通知を受信した場合、タイムアウト時間をリセットし、新たにタイムアウト時間の計測を始めるようにしたことを特徴とする第１の通信機器。

【請求項２０】　暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信

した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

第２の通信機器が通知する、公開値の生成と共有鍵の生成のために自身に必要な時間を受信し、

公開値の生成と共有鍵の生成のために自身に必要な時間と、通知された前記第２の通信機器が公開値の生成と共有鍵の生成に必要な時間とから、共有鍵交換処理を開始すべき時刻を算定し、

前記共有鍵交換処理を開始すべき時刻までに、共有鍵交換処理を開始することを特徴とする第１の通信機器。

【請求項２１】　暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

第２の通信機器が、公開値の生成と共有鍵の生成のために第２の通信機器自身に必要な時間を算出し、前記必要な時間から、共有鍵交換処理を開始すべき時刻を算定し、前記共有鍵交換処理を開始すべき時刻までに、前記第２の通信機器から、共有鍵交換処理の開始を要求された場合、共有鍵交換処理を開始することを特徴とする第１の通信機器。

【請求項２２】　前記共有鍵交換処理を開始すべき時刻は、共有鍵の生成を完了すべき時刻よりも、前記公開値の生成と共有鍵の生成のために第２の通信機器自身に必要な時間の２倍以上前であることを特徴とする請求項２１に記載の第１の通信機器。

【請求項２３】　暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

第１の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算出し、前記必要な時間から、自身が共有鍵交換処理を開始すべき時刻を算定し、

第２の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算出し、前記必要な時間から、自身が共有鍵交換処理を開始すべき時刻を算定し、

第１の通信機器と第２の通信機器のうち、前記共有鍵交換処理を開始すべき時刻が早い方が、遅い方に共有鍵交換処理の開始を通知し、共有鍵交換処理の開始を通知された場合、共有鍵交換処理を開始することを特徴とする第１の通信機器。

【請求項２４】　暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

自身が生成した第１の秘密情報を基に第１の公開値を計算し、前記第２の通信機器に前記第１の公開値を送信し、

第２の通信機器において、第２の通信機器が受信した第１の公開値が、すでに存在する鍵を更新するための要求か否かを判定し、鍵を更新するための要求と判定された場合には、第２の通信機器自身が生成した第２の秘密情報を基に前記第２の公開値を計算する処理手順を小単位に分割し、さらに小単位の処理を時間的に負荷分散させて実行することにより第２の通信機器において生成された、前記第２の公開値を受信することを特徴とする第１の通信機器。

【請求項２５】　前記第２の通信機器において、第２の公開値を時間的に負荷分散させて算出する際の処理時間を事前に算出したものを受信し、

前記第２の通信機器から通知された前記処理時間を基に、鍵の更新のために共有鍵交換の処理を開始するタイミングを算出し、前記タイミングの時点までに共有鍵交換を開始することを特徴とする、請求項２４に記載の第１の通信機器。

【請求項２６】　暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

第２の通信機器において、次回の共有鍵交換処理が、既に存在する鍵を更新す

るための処理か否かを判定し、鍵を更新するための処理と判定した場合には、第２の通信機器自身が第２の秘密情報を基に第２の公開値を計算する処理手順と共有鍵を生成する処理手順とを、小単位に分割し、さらに小単位の処理を時間的に負荷分散させて実行し、第２の通信機器において生成された前記第２の公開値を受信することを特徴とする第１の通信機器。

【請求項２７】　暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

相手から共有鍵の演算終了の通知を受けるまで、タイムアウト処理を行なわないようにすることを特徴とする第１の通信機器。

【請求項２８】　暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

第２の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算定し、

第１の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算定し、

上記、必要な時間の少なくとも一方の値を基に、共有鍵の寿命の値を決定することを特徴とする第１の通信機器。

【請求項２９】　暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

第２の公開値を送信するまでの時間が遅くなることを応答予告として第１の通信機器に対して通知することを特徴とする第２の通信機器。

【請求項３０】　暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信

した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

第１の通信機器において生成した第１の公開値を受信し、

第２の公開値を送信するまでの遅延時間または送信予定時刻を応答予告として、前記第１の公開値の受信前または受信直後に、前記第１の通信機器に予め通知しておき、その後、自身が生成した第２の秘密情報を基に前記第２の公開値を計算し、第１の通信機器に前記第２の公開値を送信し、

第１の通信機器に、今後送る予定である前記第２の公開値の送信を、少なくとも前記応答予告に基づく時間以上待たせることを特徴とする第２の通信機器。

【請求項３１】　前記第１の通信機器が、共有鍵交換の手順に先だって、前記第２の通信機器に前記応答予告を問い合わせるメッセージに対する応答メッセージとして第２公開値を送信するまでの遅延時間または送信予定時刻を前記応答予告として前記第１の通信機器に送信することを特徴とする、請求項２９または請求項３０に記載の第２の通信機器。

【請求項３２】　前記第２の通信機器は、第２の公開値を算出するための処理時間を事前に実測しておき、前記第１の通信機器に前記応答予告を通知する際には、記録しておいた実測の処理時間を基に前記応答予告である遅延時間または送信予定時刻時間を算出することを特徴とする、請求項３０または請求項３１に記載の第２の通信機器。

【請求項３３】　暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

第１の通信機器が生成した第１の公開値を受信し、

第１の通信機器が、タイムアウト時間より小さい再送間隔時間毎に、行なう状況確認を受信し、状況確認に対して、後で送信する旨の通知を、第１の通信機器に行ない、

第１の通信機器に、前記後で送信する旨の通知により、タイムアウト時間をリセットさせ、新たにタイムアウト時間の計測を始めさせるようにしたことを特徴

とする第２の通信機器。

【請求項３４】　暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

公開値の生成と共有鍵の生成のために自身に必要な時間を前記第１の通信機器に通知し、

第１の通信機器に、公開値の生成と共有鍵の生成のために第１の通信機器自身に必要な時間と、通知した前記公開値の生成と共有鍵の生成に必要な時間とから、共有鍵交換処理を開始すべき時刻を算定させ、

前記第１の通信機器に、前記共有鍵交換処理を開始すべき時刻までに、共有鍵交換処理を開始させることを特徴とする第２の通信機器。

【請求項３５】　暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

公開値の生成と共有鍵の生成のために自身に必要な時間を算出し、

前記必要な時間から、共有鍵交換処理を開始すべき時刻を算定し、

前記共有鍵交換処理を開始すべき時刻までに、第１の通信機器に、共有鍵交換処理の開始を要求することを特徴とする第２の通信機器。

【請求項３６】　前記共有鍵交換処理を開始すべき時刻は、共有鍵の生成を完了すべき時刻よりも、前記公開値の生成と共有鍵の生成のために自身に必要な時間の２倍以上前であることを特徴とする請求項３５に記載の第２の通信機器。

【請求項３７】　暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

第１の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算出し、前記必要な時間から、自身が共有鍵交換処理を開始すべき時刻を算定し

、

第２の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算出し、前記必要な時間から、自身が共有鍵交換処理を開始すべき時刻を算定し、

第１の通信機器と第２の通信機器のうち、前記共有鍵交換処理を開始すべき時刻が早い方が、遅い方に共有鍵交換処理の開始を通知し、

通知された場合、共有鍵交換処理を開始することを特徴とする第２の通信機器。

【請求項３８】　暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

第１の通信機器により生成された第１の公開値を受信し

第１の通信機器から送られた第１の公開値が、既に存在する鍵を更新するための要求か否かを判定し、

鍵を更新するための要求と判定した場合には、自分が生成した第２の秘密情報を基に前記第２の公開値を計算する処理手順を小単位に分割し、さらに小単位の処理を時間的に負荷分散させて実行し、前記第１の通信機器に前記第２の公開値を送信することを特徴とする第２の通信機器。

【請求項３９】　前記第２の通信機器は、第２の公開値を時間的に負荷分散させて算出する際の処理時間を事前に算出して、前記第１の通信機器に通知しておき、

前記第１の通信機器に、前記第２の通信機器から通知された前記処理時間を基に、鍵の更新のために共有鍵交換の処理を開始するタイミングを算出させ、前記タイミングの時点までに共有鍵交換を開始させることを特徴とする、請求項３８に記載の第２の通信機器。

【請求項４０】　暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法

に使用する通信機器であって、

次回の共有鍵交換処理が、既に存在する鍵を更新するための処理か否かを判定し、

鍵を更新するための処理と判定した場合には、自身が第２の秘密情報を基に前記第２の公開値を計算する処理手順と共有鍵を生成する処理手順とを、小単位に分割し、さらに小単位の処理を時間的に負荷分散させて実行し、第１の通信機器に前記第２の公開値を送信することを特徴とする第２の通信機器。

【請求項４１】　暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

相手から共有鍵の演算終了の通知を受けるまで、タイムアウト処理を行なわないようにすることを特徴とする第２の通信機器。

【請求項４２】　暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

第２の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算定し、

第１の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算定し、

上記、必要な時間の少なくとも一方の値を基に、共有鍵の寿命の値を決定することを特徴とする第２の通信機器。

【請求項４３】　請求項１～１２に記載の処理を、コンピュータに機能させるためのプログラムとして記録した共有鍵交換方法プログラム記録媒体。

【請求項４４】　請求項１～１２に記載の処理を、コンピュータに機能させるためのプログラムとした共有鍵交換方法のプログラム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、ネットワークを介して秘匿性の高いデータを送受信する機器間で、暗号および認証のための秘密鍵を交換・共有するための共有鍵交換方法の技術分野に関する。

【０００２】

【従来の技術】

近年のインターネットの普及により、電子メール、電子商取引など、ネットワークを介したサービスも増大し、家庭内にもネットワークに接続できる機器が増えつつある。このネットワークに接続できる家電、いわゆる「ネット家電」としては、エアコンや電子レンジなどの白物家電もインターネットに接続し、宅外からエアコンを制御したり、センター側からプログラムのバージョンアップを行うようなサービスも提案されている。

【０００３】

しかしながら、このようなネットワークの拡大に伴い、ネットワーク上の電子化されたデータを改ざんしたり、盗聴／盗み見したり、他人になりすましてサービスを受けたりする、といった問題が浮上しており、ネットワーク上のセキュリティ対策が重要となってきている。

【０００４】

ネットワーク上のセキュリティ機能を実現する取り組みとしては、暗号化技術や認証技術を使ったプロトコルが、複数、規格化・実用化されている。例えば、ＩＰパケットのレベルでのセキュリティ機能を規定したプロトコルとしては、ＩＰｓｅｃ（ＩＰ　Ｓｅｃｕｒｉｔｙ　ｐｒｏｔｏｃｏｌ）がある。

【０００５】

ＩＰｓｅｃとは、ＩＥＴＦによって標準化されたプロトコルであり、内容は、非特許文献１（ＲＦＣ２４０１）などによって規定されている。ＩＰｓｅｃでは、ＩＰパケットそのものを暗号化することで盗聴者に対して通信内容を保護し、秘密の漏洩を防ぐことができる。また、ＩＰパケットに認証（完全性チェック）用の値を追加することで、通信経路でデータが改ざんされなかったことを保証する。

【０００６】

一方、ＩＰｓｅｃなどのプロトコルを使って、機器間で暗号・認証処理されたデータを送受信する場合、データ送受信に先だって、暗号・認証処理に用いる鍵（以下、セッション共有鍵と呼ぶ）を両方の機器で共有する必要がある。ただし、ユーザが手動でセッション共有鍵を設定するのは煩雑であり、多くの機器間で通信するシステムには向かない。さらにセッション共有鍵を長時間、大量のデータに対して使いつづけると、第三者にセッション共有鍵を解読されてしまう可能性が高まるため、必要に応じて鍵を更新する必要がある。そのため、鍵を自動的に生成、配布、交換、更新するためのさまざまな方法が提案されている。

【０００７】

例えば、２つの機器間で１つの秘密の対称型鍵を交換させる方法として、広く使われている基本技術に、Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎ（ディフィーヘルマン）の方法がある。この方法は、前述のＩＰｓｅｃの鍵交換・管理プロトコルとして、標準使用が規定されているＩＫＥ（Ｉｎｔｅｒｎｅｔ　Ｋｅｙ　Ｅｘｃｈａｎｇｅ）にも使われている。非特許文献２（ＲＦＣ２４０７～２４０９）に開示されているＩＫＥの概要は、後述する。これらの鍵交換方法は、いかに安全に（第三者に鍵を推察されずに）、かつ、通信をしたい機器同士がいかに簡単に鍵を交換／共有するか、が課題である。

【０００８】

以下、従来の共有鍵交換（共有、配送）方法の具体例として、Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎの鍵交換方法を説明する。

【０００９】

図１１は、非特許文献３（ＲＦＣ２６３１）で規定されているＤｉｆｆｉｅ－Ｈｅｌｌｍａｎの鍵交換方法を説明するための図である。なお、図中で、Ｓ付きの数字は、処理のステップを表している。図中で機器Ａと機器Ｂは、暗号通信と鍵交換を行う機器の組とする。また、機器Ａと機器Ｂは鍵交換を行う前に、ｇとｎという変数の値を知っているものとする（ステップ１１０１、ステップ１１０２）。ただし、ｇとｎの値は、第三者（盗聴者）が知っていても良い。

【００１０】

まず、機器Ａは秘密の値ａを生成し（ステップ１１０３）、秘密値ａを元に公開値Ｘを算出する（ステップ１１０４）。公開値Ｘは以下の計算式で求める。

【００１１】

【数１】

$$X = g \hat{} a \mod n$$

【００１２】

ここで、上式は、ｇのａ乗をｎで割った余りを公開値Ｘとすることを意味する。算出した公開値Ｘは、機器Ｂに送る（ステップ１１０５）。

【００１３】

一方、機器Ｂは機器Ａから公開値Ｘを受け取ると、（ステップ１１０３）、（ステップ１１０４）と同様に、機器Ｂ自身の秘密値ｂの生成（ステップ１１０６）、公開値Ｙの算出（ステップ１１０７）を行う。

【００１４】

【数２】

$$Y = g \hat{} b \mod n$$

【００１５】

さらに機器Ｂでは機器Ａからの公開値Ｘと機器Ｂ自身の秘密値ｂを元に、以下の式で共有鍵Ｋを算出する（ステップ１１０８）。

【００１６】

【数３】

$$K = X \hat{} b \mod n$$

【００１７】

最後に機器Ｂは、自分の公開値Ｙを機器Ａに送る（ステップ１１０９）。機器Ａでは、機器Ｂから公開値Ｙを受け取ると、機器Ｂからの公開値Ｙと機器Ａ自身の秘密値ａを元に、以下の式で共有鍵Ｋを算出する（ステップ１１１０）。

【００１８】

【数４】

$$K = Y \hat{} a \mod n$$

【００１９】

（数３）、（数４）では、どちらの式も（数５）の関係が成立する。

【００２０】

【数５】

$$K=X^{b} \bmod n=g^{(a\times b)} \bmod n$$

$$K=Y^{a} \bmod n=g^{(a\times b)} \bmod n$$

【００２１】

そのため、機器Ａ、機器Ｂは、共有鍵Ｋとして同じ値を算出することができる。両方の機器で共有鍵Ｋを算出する、すなわち共有鍵Ｋを共有した後には、共有鍵Ｋを用いて暗号／認証処理したデータを、機器間で送受信することができる（ステップ１１１１）。

【００２２】

一方、盗聴者が既知のｇ、ｎ、および公開値Ｘ、Ｙを基に共有鍵Ｋを見つけ出すことは、いわゆる離散対数問題といわれ、大変難しい問題であると知られている。特にｎを極めて大きく（数百ビットから数千ビット）に設定した場合には、離散対数を解くことは事実上不可能である。

【００２３】

以上のように、Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎの鍵交換の手順を使うことにより、機器Ａ－機器Ｂ間で安全に共有鍵Ｋを共有させることができる。

【００２４】

なお、前述したＩＫＥの場合には、Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎの鍵交換の手順だけでなく、その前後に機器間でやりとりする手順も規定されている。以下、図１２を使ってＩＫＥでの鍵交換方法を簡単に説明する。なおＩＫＥにおいては、最初に鍵交換の要求を相手に送る側の機器をイニシエータと呼び、その要求を受け付ける側の機器をレスポンダと呼ぶ。図中の左側の処理フローはイニシエータ側の機器に対応し、右側の処理フローはレスポンダ側の機器に対応する。また、ＩＫＥではＰｈａｓｅ１（ステップ１２０１）とＰｈａｓｅ２（ステップ１２０５）という２段階の鍵交換を規定している。

【００２５】

まずＰｈａｓｅ１の段階（ステップ１２０１）において、イニシエータとレス

ポンダは、はじめに鍵交換と暗号／認証処理に用いる各種パラメータを折衝する（ステップ１２０２）。例えば、Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎで用いるｇとｎの値や、後の（ステップ１２０４）で用いる暗号化および認証のアルゴリズムもここで決められる。ｇとｎの値については、値そのもの、あるいは、ｇとｎの値を特定するグループ番号の形で取り決める。次に、前述したＤｉｆｆｉｅ－Ｈｅｌｌｍａｎの鍵交換を行い、共有鍵Ｋ＿Ｐｈａｓｅ１を共有する（ステップ１２０３）。Ｐｈａｓｅ１の最後には、イニシエータとレスポンダが相互に本当に正しい相手かどうかを認証するための手順を行う（ステップ１２０４）。このステップ１２０４の本人性認証で使うメッセージは、共有鍵Ｋ＿Ｐｈａｓｅ１に基づく鍵によって暗号／認証処理した後、送受信する。お互いの機器を認証するための情報としては、ＩＤやハッシュ値などを使うことができる。たとえば、共有鍵Ｋ＿Ｐｈａｓｅ１とステップ１２０３で交換したデータなどを連結し、それを鍵つきハッシュ関数の入力値として計算したハッシュ値を、ステップ１２０４のメッセージの暗号／認証に使う鍵として使用できる。

【００２６】

次のＰｈａｓｅ２の段階（ステップ１２０５）は、イニシエータ・レスポンダ間で実際に送受信したいデータに対する暗号化／認証処理について決定する場である。Ｐｈａｓｅ２のメッセージでは、ステップ１２０４と同様、共有鍵Ｋ＿Ｐｈａｓｅ１に基づく鍵で暗号化および認証処理が施される。Ｐｈａｓｅ２の最初の２メッセージでは、ｇ、ｎの値の提案又は通知、暗号化および認証のアルゴリズムなどの折衝と、Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎの公開値の交換を行う（ステップ１２０６）。Ｐｈａｓｅ２の（ｇ、ｎ、ａ、ｂ）には、Ｐｈａｓｅ１のパラメータと異なる値を使うことができる。イニシエータとレスポンダは、ステップ１２０６のＤｉｆｆｉｅ－Ｈｅｌｌｍａｎ交換で計算される共有鍵Ｋ＿Ｐｈａｓｅ２を基に、実際のデータに適用する暗号・認証処理の鍵を生成し、暗号化・認証処理済みのデータを送受信できるようになる（ステップ１２０７）。

【００２７】

ＩＫＥでは、上記手順以外にも、一方のメッセージが他方の機器に届かなかった場合の対策であるメッセージの再送や、鍵の更新手順なども規定されている。

【００２８】

【非特許文献１】

「ＲＦＣ２４０１」、ＩＥＴＦ（Ｉｎｔｅｒｎｅｔ　Ｅｎｇｉｎｅｅｒｉｎｇ　Ｔａｓｋ　Ｆｏｒｃｅ）発行

【非特許文献２】

「ＲＦＣ２４０７」～「ＲＦＣ２４０９」、ＩＥＴＦ（Ｉｎｔｅｒｎｅｔ　Ｅｎｇｉｎｅｅｒｉｎｇ　Ｔａｓｋ　Ｆｏｒｃｅ）発行

【非特許文献３】

「ＲＦＣ２６３１」、ＩＥＴＦ（Ｉｎｔｅｒｎｅｔ　Ｅｎｇｉｎｅｅｒｉｎｇ　Ｔａｓｋ　Ｆｏｒｃｅ）発行

【００２９】

【発明が解決しようとする課題】

上記のように、従来提案されてきたＤｉｆｆｉｅ－Ｈｅｌｌｍａｎなどの共有鍵交換法を使うことで、通信を行う機器間で鍵を安全に共有させることができる。

【００３０】

しかしながら、Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎの共有鍵交換法はべき乗演算とモジュロ演算を含んでおり、桁数の大きい数値を使う場合の処理負荷は非常に重いことが知られている。例えば、ＩＫＥで規定されているＤＨグループ２の場合には、ｇ＝２、ｎ：１０２４ビットの定数を使って公開値Ｘ、Ｙおよびセッション共有鍵Ｋを計算する。特にネット家電のように低コストのＣＰＵを搭載する機器では、高い計算力が望めず、以下のような３つの課題が発生する。

【００３１】

・第１の課題：対向の機器でのタイムアウト検出により、鍵交換が失敗する。

【００３２】

・第２の課題：鍵の寿命が設定されている場合、寿命満了までに、鍵の更新が完了しない。

【００３３】

・第３の課題：機器上の他のアプリケーションの実行を妨害する。

【００３４】

（第１の課題）対向の機器でのタイムアウト検出による鍵交換の失敗

以下、図１３を使って第１の課題が発生する具体例を説明する。図１３は、機器Ａと機器Ｂの間でＤｉｆｆｉｅ－Ｈｅｌｌｍａｎの鍵交換を行うシーケンスを示したものであり、機器Ｂは低い性能の（計算力の低い）ＣＰＵを搭載する機器とする。

【００３５】

まず、図１３のステップ１３０１～１３０６は、図１１の１１０１～１１０６と同じであり、機器Ａ側からＤｉｆｆｉｅ－Ｈｅｌｌｍａｎの公開値Ｘを送る。次に、機器Ｂは図１１と同様に公開値Ｙとセッション共有鍵Ｋの演算を行うが（ステップ１３０７、１３０８）、ＣＰＵの計算力が低いため、非常に長い処理時間を要する。

【００３６】

機器Ａ側では、自分の公開値Ｘを送った後（ステップ１３０５）、対向の機器（図１３では機器Ｂ）からの公開値Ｙの応答を待つが、通常、通信経路中のパケットロスや対向機器の障害などの事態に備えて、再送やタイムアウトの機構を持たせている。図１３の機器Ａの場合は、機器Ｂからの応答が無ければ再送間隔＝Ｔ［秒］の時間毎に公開値Ｘのメッセージを再送する（ステップ１３１０）。さらに再送を３回繰り返し（ステップ１３１１、１３１２）、それでも機器Ｂからの応答が無い場合には、Ｔ×４［秒］後にタイムアウトを検出して、鍵交換が失敗したとみなす（ステップ１３１３）。

【００３７】

その後、聞きＢで公開値Ｙとセッション共有鍵Ｋの演算が終了し、公開値Ｙを送った（ステップ１３０９）としても、そのときには機器Ａでは既に機器Ｂとの鍵交換処理中の状態をリセットしている。そのため、公開値Ｙを含むメッセージは、無効なメッセージと判断して廃棄されてしまう（ステップ１３１４）。

【００３８】

（第２の課題）鍵更新の完了前に、古い鍵の寿命が切れる課題

同一の鍵を長時間にわたって使用すると、その間に鍵を解読されることが全く

ないとは言えない。そこで、ＩＫＥでは、鍵に寿命を設けておき、適当な時間が経つと、それまでの鍵の使用を終了し、別の新しい鍵を使用することができる。具体的には、鍵交換成功により、鍵が作成されると、その鍵に寿命を設定する。あるいは、鍵交換の手順を開始する際に、その鍵に付与する寿命満了の時刻を機器Ａ、Ｂ間で決めておく。その鍵の寿命が満了になる前に、次の新しい鍵を作成し、使用中の鍵の寿命が満了となる前に、この新しい鍵に切り替える。このような鍵の更新処理を、リキー処理と呼ぶ。Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎに伴う演算に時間がかかると、リキー処理が遅れ、古い鍵の寿命満了までに鍵の更新ができず、古い鍵を使い続けるうちに、その寿命が満了するとパケットの暗号化ができなくなる課題があった。

【００３９】

（第３の課題）他のアプリケーションへの妨害

Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎに伴う演算を行うと、長時間にわたってネット家電のＣＰＵをこの演算だけで占有してしまい、ＣＰＵリソースが枯渇する。そのため、ネット家電上で動作する他のアプリケーションの実行が妨げられ、正常に動作しなくなる場合がある。

【００４０】

それゆえ、本発明の第１の目的は、低性能のＣＰＵを使う場合にも、対向の機器でタイムアウトを検出させずに鍵交換を成功させるような共有鍵交換方法を提供することである。本発明の第２の目的は、低性能のＣＰＵを使う場合にも、リキーが正常に行なわれるような共有鍵交換方法を提供することである。さらに本発明の第３の目的は、低性能のＣＰＵを使う場合にも、同じＣＰＵ上の他のアプリケーションの実行を妨げることなく、同時に鍵交換も成功させるような共有鍵交換方法を提供することである。

【００４１】

【課題を解決するための手段】

本発明では、上記課題を解決するために、以下のような、方法、および、手段を適用する。

【００４２】

暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、前記２つの通信機器は、第１の通信機器と第２の通信機器から成り、前記第１の通信機器および前記第２の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、前記第１の通信機器は、自分が生成した第１の秘密情報を基に第１の公開値を計算し、前記第２の通信機器に前記第１の公開値を送信し、前記第２の通信機器は、第２の公開値を送信するまでの時間が遅くなることを応答予告として、前記第１の公開値の受信前または受信直後に、前記第１の通信機器に予め通知しておき、その後、自分が生成した第２の秘密情報を基に前記第２の公開値を計算し、前記第１の通信機器に前記第２の公開値を送信し、前記第１の通信機器は、前記第２の通信機器から前記応答予告を受け取るようにしたことを特徴とする共有鍵交換方法。

【００４３】

暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、前記２つの通信機器は、第１の通信機器と第２の通信機器から成り、前記第１の通信機器および前記第２の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、前記第１の通信機器は、自分が生成した第１の秘密情報を基に第１の公開値を計算し、前記第２の通信機器に前記第１の公開値を送信し、前記第２の通信機器は、第２の公開値を送信するまでの遅延時間又は送信予定時刻を応答予告時間として、前記第１の公開値の受信前または受信直後に、前記第１の通信機器に予め通知しておき、その後、自分が生成した第２の秘密情報を基に前記第２の公開値を計算し、前記第１の通信機器に前記第２の公開値を送信し、前記第１の通信機器は、前記第２の通信機器から前記応答予告時間を受け取ると、前記第２の通信機器から今後送られてくるはずの前記第２の公開値を、少なくとも前記応答予告に基づく時間以上待機することを特徴とする共有鍵交換方法。

【００４４】

前記第１の通信機器は、共有鍵交換の手順に先だって、前記第２の通信機器に

前記応答予告時間を問い合わせるメッセージを発行し、前記第２の通信機器は前記メッセージに対して、第２の公開値を送信するまでの遅延時間または送信予定時刻を応答予告として、前記第１の通信機器に送信することを特徴とする、請求項１または２に記載の共有鍵交換方法。

【００４５】

前記第２の通信機器は、第２の公開値を算出するための処理時間を事前に実測しておき、前記第１の通信機器に前記応答予告を通知する際には、記録しておいた実測の処理時間を基に前記応答予告である遅延時間または送信予定時刻を算出することを特徴とする、請求項２または３に記載の共有鍵交換方法。

【００４６】

暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、前記２つの通信機器は、第１の通信機器と第２の通信機器から成り、前記第１の通信機器および前記第２の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、前記第１の通信機器は、自分が生成した第１の秘密情報を基に第１の公開値を計算し、前記第２の通信機器に前記第１の公開値を送信し、前記第１の通信機器は、タイムアウト時間より小さい再送間隔時間毎に、第２の通信機器に対して状況確認を行ない、前記第２の通信機器は、後で送信する旨の通知を、前記第１の通信機器に行ない、前記第１の通信機器は、後で送信する旨の通知を受信した場合、タイムアウト時間をリセットし、新たにタイムアウト時間の計測を始めるようにしたことを特徴とする共有鍵交換方法。

【００４７】

暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、前記２つの通信機器は、第１の通信機器と第２の通信機器から成り、前記第１の通信機器および前記第２の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、前記第２の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を前記第１の通信機器に通知し、前記第１の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間と、通知された

前記第２の通信機器が公開値の生成と共有鍵の生成に必要な時間とから、共有鍵交換処理を開始すべき時刻を算定し、前記第１の通信機器は、前記共有鍵交換処理を開始すべき時刻までに、共有鍵交換処理を開始することを特徴とする共有鍵交換方法。

【００４８】

暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、前記２つの通信機器は、第１の通信機器と第２の通信機器から成り、前記第１の通信機器および前記第２の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、前記第２の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算出し、前記第２の通信機器は、前記必要な時間から、共有鍵交換処理を開始すべき時刻を算定し、前記第２の通信機器は、前記共有鍵交換処理を開始すべき時刻までに、前記第１の通信機器に、共有鍵交換処理の開始を要求することを特徴とする共有鍵交換方法。

【００４９】

前記共有鍵交換処理を開始すべき時刻は、共有鍵の生成を完了すべき時刻よりも、前記公開値の生成と共有鍵の生成のために自身に必要な時間の２倍以上前であることを特徴とする請求項７に記載の共有鍵交換方法。

【００５０】

暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、前記２つの通信機器は、第１の通信機器と第２の通信機器から成り、前記第１の通信機器および前記第２の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、前記第１の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算出し、前記必要な時間から、自身が共有鍵交換処理を開始すべき時刻を算定し、前記第２の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算出し、前記必要な時間から、自身が共有鍵交換処理を開始すべき時刻を算定し、前記第１の通信機器と第２の通信機器のうち、前記共有鍵交換処理を開始すべき時刻が早い方が、遅い方に共有鍵交換処

理を開始し、前記第１の通信機器と第２の通信機器は、共有鍵交換処理を開始することを特徴とする共有鍵交換方法。

【００５１】

暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、前記２つの通信機器は、第１の通信機器と第２の通信機器から成り、前記第１の通信機器および前記第２の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、前記第１の通信機器は、自分が生成した第１の秘密情報を基に第１の公開値を計算し、前記第２の通信機器に前記第１の公開値を送信し、前記第２の通信機器は、前記第１の通信機器から送られた第１の公開値が、既に存在する鍵を更新するための要求か否かを判定し、鍵を更新するための要求と判定した場合には、自分が生成した第２の秘密情報を基に前記第２の公開値を計算する処理手順を小単位に分割し、さらに小単位の処理を時間的に負荷分散させて実行し、前記第１の通信機器に前記第２の公開値を送信することを特徴とする共有鍵交換方法。

【００５２】

前記第２の通信機器は、第２の公開値を時間的に負荷分散させて算出する際の処理時間を事前に算出して、前記第１の通信機器に通知しておき、前記第１の通信機器は、前記第２の通信機器から通知された前記処理時間を基に、鍵の更新のために共有鍵交換の処理を開始するタイミングを算出し、前記タイミングの時点までに共有鍵交換を開始することを特徴とする、請求項１０に記載の共有鍵交換方法。

【００５３】

暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、前記２つの通信機器は、第１の通信機器と第２の通信機器から成り、前記第１の通信機器および前記第２の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、前記第２の通信機器は、次回の共有鍵交換処理が、既に存在する鍵を更新するための処理か否かを判定し、鍵を更新するための処

理と判定した場合には、自身が第２の秘密情報を基に前記第２の公開値を計算する処理手順と共有鍵を生成する処理手順とを、小単位に分割し、さらに小単位の処理を時間的に負荷分散させて実行し、前記第１の通信機器に前記第２の公開値を送信することを特徴とする共有鍵交換方法。

【００５４】

暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、前記２つの通信機器は、第１の通信機器と第２の通信機器から成り、前記第１の通信機器および前記第２の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、前記第１、または、第２の通信機器は、共有鍵の演算を終了後、演算終了の通知を、相手の通信機器に行ない、相手の通信機器は、演算終了の通知を受けるまで、タイムアウト処理を行なわないようにすることを特徴とする共有鍵交換方法。

【００５５】

暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、前記２つの通信機器は、第１の通信機器と第２の通信機器から成り、前記第１の通信機器および前記第２の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、前記第２の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算定し、前記第１の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算定し、上記、必要な時間の少なくとも一方の値を基に、共有鍵の寿命の値を決定することを特徴とする共有鍵交換方法。

【００５６】

上記請求項１～１４に記載の処理を行なう、上記第１の通信機器、または、第２の通信機器。

【００５７】

上記請求項１～１４に記載の処理を、コンピュータに機能させるためのプログラムを記録した共有鍵交換方法プログラム記録媒体。

【００５８】

上記請求項１～１４に記載の処理を、コンピュータに機能させるためのプログラムとした共有鍵交換方法のプログラム。

【００５９】

【発明の実施の形態】

図２は、本発明の共有鍵交換方法が適用されるネットワーク構成を示す図である。本発明の共有鍵交換方法が適用される実装形態は、ルータやゲートウェイ（ＧＷ）の機器に実装されるケース（図２（ａ））と、端末に実装されるケース（図２（ｂ））の２種類に分類できる。

【００６０】

まず、図２（ａ）は、ＧＷ型機器Ａ２０１およびＧＷ型機器Ｂ２０２に本発明の共有鍵交換処理が実装されるケースを示す。図２（ａ）において、ＧＷ型機器Ａ２０１とＧＷ型機器Ｂ２０２は、公衆網２０７を介して接続されている。また、ＧＷ型機器Ａ２０１は、自身のＬＡＮ２０８内の端末である端末Ａ－１　２０３、端末Ａ－２　２０４とも接続されている。同様にＧＷ型機器Ｂ２０２は、自身のＬＡＮ２０９内の端末である端末Ｂ－１　２０５、端末Ｂ－２　２０６とも接続されている。

【００６１】

ＬＡＮ２０８内の端末とＬＡＮ２０９内の端末の間は、ＧＷ型機器Ａ２０１とＧＷ型機器Ｂ２０２を中継して通信でき、ＧＷ型機器Ａ２０１とＧＷ型機器Ｂ２０２との間のデータは暗号化・認証処理される。すなわち、暗号化・認証処理は２つのＧＷ型機器間で終端しており、２つのＧＷ型機器は、それぞれ自身のＬＡＮ端末（ＬＡＮ２０８内の機器とＬＡＮ２０９内の機器）からのデータを暗号化・認証処理して対向のＧＷに送信し、対向のＧＷからの暗号化・認証処理後のデータを受信すると、復号化・認証処理した後それぞれ自身のＬＡＮ端末にフォワードする。また、共有鍵の交換処理もＧＷ型機器Ａ２０１とＧＷ型機器Ｂ２０２の間で終端する。

【００６２】

一方、図２（ｂ）では、ホスト型機器Ａ２１０およびホスト型機器Ｂ２１１に

本発明の共有鍵交換処理が実装されるケースを示す。図２（ｂ）において、ホスト型機器Ｂ２１１は、公衆網２０７を介してＧＷ型機器Ａ２０１またはホスト型機器Ａ２１０と接続されている。ＧＷ型機器Ａ２０１と自身のＬＡＮ２０８、ＬＡＮ内の端末Ａ－１　２０３、端末Ａ－２　２０４は、図２（ａ）と同じである。

【００６３】

ホスト型機器ＢとＬＡＮ２０８内の端末との間は、ＧＷ型機器Ａ２０１を中継して通信でき、ホスト型機器Ｂ２１１とＧＷ型機器Ａ２０１と間のデータは暗号化・認証処理される。すなわち、ホスト型機器Ｂ２１１は、ＧＷ型機器Ａ２０１と異なり、自身内部の発着データを自分で暗号化・認証処理する。さらにホスト型機器Ｂ２１１は、同様の機能を持つホスト型機器Ａ２１０との間で、暗号化・認証処理済みのデータを通信できる。また、共有鍵の交換処理に関しては、ホスト型機器Ｂ２１１－ＧＷ型機器Ａ２０１の間、およびホスト型機器Ｂ２１１－ホスト型機器Ａ２１０の間で、それぞれ、独立で終端する。

【００６４】

次に、本発明の共有鍵交換方法が適用される機器の機能ブロックについて図３を使って説明する。図３は、図２で説明したＧＷ型機器Ｂ２０２の機能ブロックを示す。図３のＧＷ型機器Ｂ２０２において、共有鍵交換部３０１、データベース管理部３０２、データベース部３０３、暗号化／認証処理部３０４、通信プロトコル処理部３０５、３０６、ＬＡＮ　Ｉ／Ｆ３０７、ＷＡＮ　Ｉ／Ｆ３０８とを備える。

【００６５】

ＧＷ型機器Ｂ２０２は、ＷＡＮ　Ｉ／Ｆ３０８を介して公衆網に接続しており、ＧＷ型機器Ａ２０１と通信できる。一方、ＬＡＮ２０９とはＬＡＮ　Ｉ／Ｆ３０７を介して接続しており、ＬＡＮ内の端末２０５、２０６と通信できる。通信プロトコル処理部３０５、３０６は、ＩＰレイヤ、ＴＣＰレイヤなどの通信プロトコルを処理する部分であり、ＬＡＮ内の端末と公衆網を介した機器との間の送受信パケットのルーティング処理なども行う。

【００６６】

共有鍵交換部３０１は、本発明の共有鍵交換方法に基づく処理を行う処理部である。共有鍵交換部３０１では、対向の機器であるＧＷ型機器Ａ２０１と鍵交換用の通信を行い、ＧＷ型機器Ａ２０１と共有鍵を共有する。共有鍵は、データベース管理部３０２を介してデータベース部３０３に記録される。

【００６７】

データベース部３０３は、暗号／認証処理に用いる共有鍵を保存する記録部であり、鍵だけでなく、暗号アルゴリズムなどの情報を保存する。データベース部３０３への情報の登録、削除などの処理は、データベース管理部３０２が行う。暗号化／認証処理部３０４は、データベース部３０３の鍵情報等を参照して、パケットの暗号化・復号化／認証（完全性チェック）の処理を行う。具体的には、ＬＡＮ２０９内の端末Ｂ－１　２０５、または、端末Ｂ－２　２０６からＬＡＮ２０９内の端末Ａ－１　２０３、または、端末Ａ－２　２０４への送信パケットは、平文（暗号化されていない通常のパケット）で受け取る。ＬＡＮ　Ｉ／Ｆ３０７、通信プロトコル処理部３０５を介して暗号化／認証処理部３０４が上記パケットを受け取ると、データを暗号化、認証処理用のデータの追加、などの処理を行い、通信プロトコル処理部３０６、ＷＡＮ　Ｉ／Ｆ３０８を介して公衆網に送出する。暗号化／認証処理部３０４以降の通信プロトコル処理部３０６、ＷＡＮ　Ｉ／Ｆ３０８を介して送信されるパケットは、暗号化済みとなる。また、公衆網側から到着した暗号化済みのパケットの場合も同様に、暗号化／認証処理部３０４で復号化の処理を行い、ＬＡＮ２０９内の端末に対して平文のデータを中継する。

【００６８】

（実施の形態１）

以下、本発明の実施の形態１に係る共有鍵交換方法について説明する。実施の形態１は、従来の共有鍵交換方法における３つの課題のうち、第１の課題を解決するための方法である。

【００６９】

図１は、本発明の実施の形態１に係る共有鍵交換方法の概要を説明するための図である。図１において、機器Ａと機器Ｂは、暗号・認証処理と共有鍵交換を終

端する機器の組とする。具体的には、図２中の機器間で共有鍵交換の処理を終端する３つの組み合わせ、すなわち（１）ＧＷ型機器Ａ２０１とＧＷ型機器Ｂ２０２、（２）ＧＷ型機器Ａ２０１とホスト型機器Ｂ２１１、（３）ホスト型機器Ａ２１０とホスト型機器Ｂ２１１のいずれかの組み合わせに対応するものとする。また、図１中の機器Ｂは、機器Ａに比べて低い性能のＣＰＵを搭載した機器とする。また、図１の機器Ａまたは機器Ｂに関する処理シーケンスは、図３における共有鍵交換部３０１で行う処理に対応するものとする。

【００７０】

まず、図１のステップ１０１～１０５は、図９のステップ９０１～９０５と同じであり、機器Ａは、算出したＤｉｆｆｉｅ－Ｈｅｌｌｍａｎの公開値Ｘを機器Ｂに送る。次に、機器Ｂはこれから自分が計算すべき公開値Ｙと共有鍵Ｋの演算所要時間の推定値：Ｔｂを求める（ステップ１０６）。具体的には、ｎとｇの取り得る値のさまざまな組み合わせに関して、事前に機器Ｂ上で演算を複数回実行しておき、その最大値Ｔｉｍｅ＿ｍａｘ［ｇ，ｎ］を保存しておく。この値に、予め決めておいた固定値：α＝機器Ａ／機器Ｂ間のメッセージ送受信時の遅延時間と揺らぎを考慮した値、を足して、以下の式で推定値Ｔｂを求める。

【００７１】

【数６】

$$Tb = Time\_max[g, n] + \alpha$$

【００７２】

機器Ｂは、算出したＴｂを機器Ａに送り、Ｔｂ時間以内に公開値Ｙを応答することを予告する（ステップ１０７）。機器Ａは、Ｔｂを受け取ると、機器Ｂからの応答待ち期限をＴｂ以上に延長する（ステップ１１１）。すなわち、機器Ａでは機器Ｂからの公開値ＹをＴｂの時間以上待つように制御することで、機器Ｂから最大Ｔｂの時間、応答が無くても、タイムアウトとみなさなくなる。

【００７３】

機器Ｂは、Ｔｂを機器Ａに送信後、図１１のステップ１１０６～ステップ１１０８と同様の手順で、公開値Ｙと共有鍵Ｋを求める（ステップ１０８～ステップ１１０）。公開値Ｙの計算が終わると、機器Ａに対して公開値Ｙを送る（ステッ

プ１１２）。

【００７４】

機器Ａでは、機器Ｂから公開値Ｙを受け取ると、応答待ち期限：Ｔｂが満了していないため、Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎの公開値の応答と判断して正しく処理できる。従って、機器Ａ側でも共有鍵Ｋを計算し（ステップ１１３）、機器Ａ、機器Ｂの両方で鍵の交換が成功する。

【００７５】

以上のように、低い性能のＣＰＵを搭載した機器Ｂは、鍵交換に伴う高負荷の演算の所要時間を事前に推定し、応答メッセージの最大遅延時間を対向の機器（機器Ａ）に予告しておく。これにより、従来の方法で課題であった、低い性能のＣＰＵ搭載機器がまだ鍵交換の演算をしている間に、対向の機器がタイムアウトと判断してしまうような課題が発生せず、鍵交換が正常に成功できるようになる。

【００７６】

なお、図１において機器Ｂは、相手（機器Ａ）から公開値Ｘを受信した後に、演算所要時間Ｔｂを予告するメッセージを送り返しているが、演算所要時間を相手に予告する手順は、図１の手順に限定されるものではない。

【００７７】

例えば、機器ＡがＤｉｆｆｉｅ－Ｈｅｌｌｍａｎの公開値Ｘを計算し始める前に、予め機器Ａ、機器Ｂ間でお互いの演算所要時間を通知する手順を行っても良い。具体的には、図４のステップ４０１で、機器Ａ側から所要時間の問い合わせ要求を発行し、それに対して機器Ｂが演算所要時間を推定、応答する（ステップ４０１～ステップ４０３）。ステップ４０２、ステップ４０３は、図１のステップ１０６、ステップ１０７と同様の手順である。その後の処理は、図１のステップ１０１以降の処理（ただしステップ１０６、１０７を除く）と同じである。

【００７８】

また、実施の形態１では、Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎの鍵交換時に使う演算として、離散対数問題を利用したべき乗／モジュロ演算のケースを説明したが、本発明の適用範囲はこの演算に限定されるものではない。例えば、Ｄｉｆｆｉ

ｅ－Ｈｅｌｌｍａｎの変形として、楕円曲線に基づく数学的集合を用いた演算方法を使っても良い。

【００７９】

また、Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎに関する演算所要時間の推定方法（図１のステップ１０６、図４のステップ４０２の処理）として、事前に計算した結果を保存しておく方法を説明したが、この方法に限定するものではない。例えば、その時点のＣＰＵ使用率を測定し、ＣＰＵ使用率が既に他のアプリケーション等によって使われている場合には、演算所要時間を長く推定してもよい。さらに、複数の機器との間の鍵交換が同時期に集中する可能性がある場合には、複数のＤｉｆｆｉｅ－Ｈｅｌｌｍａｎの処理がキューに溜まる可能性がある。そのため、Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎの処理待ちキューなどを設け、処理順番に応じて演算所要時間を長く推定しても良い。

【００８０】

また、実施の形態１の処理シーケンスとして図１では、Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎの演算と公開値のやりとり部分に限定して説明したが、その他のＤｉｆｆｉｅ－Ｈｅｌｌｍａｎを利用する鍵交換プロトコルに適用してもよい。例えば、図１２で説明したように、ＩＫＥのシーケンスでは２箇所でＤｉｆｆｉｅ－Ｈｅｌｌｍａｎの鍵交換を利用しているが、実施の形態１で説明したメッセージを、ＩＫＥのその他のメッセージと連携させて実行してもよい。具体的には、図４のステップ４０１とステップ４０３で行っている演算所要時間の問い合わせ・通知は、ＩＫＥのＰｈａｓｅ１の各種パラメータの折衝（図１２のステップ１２０２）の中で行っても良い。

【００８１】

また、実施の形態１では、機器Ｂにおける共有鍵Ｋの算出は、自身の公開値Ｙの算出と同時に行っていた（公開値Ｙの送信前に算出していた）が、必ずしも公開値Ｙの送信前に行う必要はない。機器Ａ側での共有鍵Ｋの算出は、公開値Ｙの受信後に行うので、むしろ、機器Ｂ側でも公開値Ｙの送信後に共有鍵Ｋを算出した方が鍵交換全体の処理時間が短く抑えられる場合がある。なお、機器Ｂにおける共有鍵Ｋの算出が、公開値Ｙの送出後に行われる場合には、実施の形態１の処

理手順において、機器Ｂでの演算所要時間の推定値：Ｔｂ（図１のステップ１０６）は、公開値Ｙの算出のみの時間として推定する必要がある。

【００８２】

なお、実施の形態１では、機器Ｂの演算所要時間だけを機器Ａに通知・報告する手順を説明したが、機器Ａと機器Ｂがお互いの演算所要時間を通知・報告し合っても良い。

【００８３】

（実施の形態２）

実施の形態１では、公開値Ｙや共有鍵Ｋの演算の所要時間を問い合わせ、通知することにより、タイムアウトの発生を防止する方法であったが、所要時間を通知しない方法を、図５を用いて、次に説明する。

【００８４】

図５において、ステップ５０１～ステップ５０４は、図１におけるステップ１０１～ステップ１０４と同様である。機器Ａは、ステップ５０５において、公開値Ｘを機器Ｂに通知する。機器Ｂは、公開値Ｙの演算を依頼されたが、後で公開値Ｙを送信することのみを機器Ａに通知し（ステップ５０６）、秘密情報ｂを選定し（ステップ５０７）、公開値Ｙの演算（ステップ５０８）と共有鍵Ｋの演算（ステップ５１１）を始める。ステップ５０６により、後で公開値Ｙが送信されることを通知された機器Ａは、再送間隔Ｔ［秒］を設定し、再送間隔Ｔが経過すると、状況確認・再送をステップ５０９において行なう。

【００８５】

なお、ステップ５０５、ステップ５０６において、パケットロスが起きることもあるので、ステップ５０５の後で、機器Ａは一定の再送間隔を設定して、再送間隔以内に機器Ｂからの通知がない場合、ステップ５０９と同様の状況確認・再送を行なうようにしてもよい。

【００８６】

機器Ｂは、公開値Ｙ、共有鍵Ｋの演算が終了していないので、ステップ５１０において、再び、後で公開値Ｙを送信することのみを機器Ａに通知する。機器Ａは、再送間隔Ｔが経過するとステップ５１２において、状況確認・再送を行なう

。機器Ｂは、まだ、公開値Ｙ、共有鍵Ｋの演算が終了していないので、ステップ５１３において、再び、後で公開値Ｙを送信することのみを機器Ａに通知する。機器Ａは、再送間隔Ｔが経過するとステップ５１４において、状況確認・再送を行なう。機器Ｂは、まだ、公開値Ｙ、共有鍵Ｋの演算が終了していないので、ステップ５１５において、再び、後で公開値Ｙを送信することのみを機器Ａに通知する。その後で、公開値Ｙ、共有鍵Ｋの演算が終了し、公開値Ｙを機器Ａに送信する（ステップ５１６）。機器Ａは、ステップ５１７において、入手した公開値Ｙと秘密値ａにより共有鍵Ｋを算出し、鍵交換が終了する。機器Ａは、機器Ｂから、ステップ５０６、ステップ５１０、ステップ５１３、ステップ５１５により、後で公開値Ｙを送信することの通知を受けたら、図７で説明したタイムアウト時間をリセットし、新たにタイムアウト時間の計測を始める。再送間隔Ｔ［秒］をタイムアウト時間より短くしておけば、タイムアウトの発生はなくなる。

【００８７】

また、機器Ｂが一旦、ステップ５０６のような　“後で公開値Ｙを送信する旨の通知”を行うと、機器Ａ側ではタイムアウト時間を設定せずに無限に待ちつづける処理に切り替えても良い。

【００８８】

なお、ステップ５１６の公開値Ｙの通知は、ステップ５０８の後に行なってもよい。この場合は、機器Ａは、公開値Ｙを受信した後は、機器Ｂに対して、共有鍵Ｋの演算が終了したかどうかの質問を、状況確認として行ない、これに対して、機器Ｂは、共有鍵Ｋの演算中であることを、機器Ａに回答し、機器Ａは、回答を受けた後、タイムアウト時間をリセットし、新たにタイムアウト時間の計測を始め、機器Ｂから共有鍵Ｋの演算終了の回答が帰ってくるまで、繰り返すようにすればよい。機器Ａのステップ５１７が終了し、機器Ｂから共有鍵Ｋの演算終了の回答を得たら、共有鍵交換が完了する。

【００８９】

現状のＩＫＥの規定では、機器Ｂが公開値Ｙを通知したときに、共有鍵Ｋの算出まで完了しているべきか否かの規定はなく、機器Ａ側で機器Ｂの共有鍵Ｋの算出完了を、タイムアウト時間を設けて待つような規定もない。従って、必ずしも

機器Ａ側でのタイムアウト計測を行う必要はないのであるが、上記のように、機器Ａ側でタイムアウト計測をしてもよい。

【００９０】

（実施の形態３）

次に本発明の実施の形態３に係る共有鍵交換方法について説明する。実施の形態３は、従来の共有鍵交換方法における３つの課題のうち、第２の課題を解決するための方法である。

【００９１】

まず、実施の形態３の概要について、図６を用いて説明する。先にも述べたように、同一に鍵を長時間にわたって使用すると、その間に鍵が第３者によって解読されることが全く無いとは言えない。そこで、鍵の作成に際して、その鍵に寿命を設定するようにし、鍵交換が成功するとその鍵には寿命が設けられているようにする。その鍵の寿命が満了になる前に、次の鍵の作成処理であるリキー処理を開始し、次の新しい鍵を作成し、使用中の鍵の寿命が満了となる前に、この新しい鍵に切り替える。実施の形態３では、リキー処理の完了時刻が、古い鍵の寿命満了時刻よりも前になるように、リキー処理の開始時刻を算出し、リキー処理を開始する方法である。

【００９２】

図７は、本発明の共有鍵交換方式におけるリキー処理の手順を示す図である。機器Ａは、ステップ７０１において、機器Ｂに、所要時間、すなわち、公開値Ｙの演算に必要な時間Ｔｂ＿ｌｏｎｇ１と共有鍵Ｋの演算に必要な時間Ｔｂ＿ｌｏｎｇ２を問い合わせる。機器Ｂは、ステップ７０３において、Ｔｂ＿ｌｏｎｇ１とＴｂ＿ｌｏｎｇ２の推定を行ない、ステップ７０４において、機器Ａに回答する。機器Ａは、ステップ７０２において、機器Ａが行なう公開値Ｘの演算に必要な時間Ｔａ＿ｌｏｎｇ１と共有鍵Ｋの演算に必要な時間Ｔａ＿ｌｏｎｇ２の推定を行ない、ステップ７０５において、ｔ＿ｓｔａｒｔを算定する。

【００９３】

【数７】

$$t\_start < t\_endA - (Ta\_long1 + Ta\_long2 + Tb\_long1 + Tb\_long2) - \alpha$$

【００９４】

αは、機器Ａ、機器Ｂ間のメッセージ送受信時の遅延時間と揺らぎを考慮した値である。

【００９５】

ここで、ｔ＿ｅｎｄＡは、古い鍵の寿命満了時刻である。Ｔａ＿ｌｏｎｇ１、Ｔａ＿ｌｏｎｇ２、Ｔｂ＿ｌｏｎｇ１、Ｔｂ＿ｌｏｎｇ２は、それぞれ、演算の正味時間に、ステップ７０６、ステップ７０７などのように、その前後に必要な付随する処理時間も含んだ値である。

【００９６】

機器Ａは、時刻ｔ＿ｓｔａｒｔが来ると、リキー処理を開始する。すなわち、ステップ７０６、ステップ７０７において、変数ｇとｎを機器Ａと機器Ｂの間で共有化し、ステップ７０８において、秘密値ａを生成し、ステップ７０９において、公開値Ｘを演算する。これらの処理に、Ｔａ＿ｌｏｎｇ１の時間がかかる。機器Ａは、ステップ７１０において、公開値Ｘを機器Ｂに通知する。機器Ｂは、ステップ７１１において、秘密値ｂを生成し、ステップ７１２において、公開値Ｙを演算し、ステップ７１３において、共有鍵Ｋを生成する。ステップ７１１、ステップ７１２の処理に、Ｔｂ＿ｌｏｎｇ１かかる。さらに、ステップ７１３の処理に、Ｔｂ＿ｌｏｎｇ２かかる。機器Ｂは、ステップ７１４において、公開値Ｙを機器Ａに通知する。機器Ａは、ステップ７１５において、共有鍵Ｋを演算して生成する。この処理に、Ｔａ＿ｌｏｎｇ２かかる。ｔ＿ｓｔａｒｔを（数７）としたので、ステップ７１５の処理は、ｔ＿ｅｎｄＡまでに完了する。すなわち、古い鍵の寿命満了以前に、新しい鍵が誕生した。以降は、この新しい鍵が使用される。

【００９７】

また、ステップ７１０のあとに機器Ａにおいて、Ｔｂ＿ｌｏｎｇ１＋Ｔｂ＿ｌｏｎｇ２＋α以上のタイムアウト時間を設定することにより、第１の課題も解決

できる。

【００９８】

ステップ７１４の、公開値Ｙの機器Ｂから機器Ａへの通知は、ステップ７１２のあとですぐに行なってもよい。この場合は、ｔ＿ｓｔａｒｔを次の式により求める。

【００９９】

【数８】

$$t\_start < t\_endA - \{Ta\_long1 + Tb\_long1 + MAX(Ta\_long2, Tb\_long2)\} - \alpha$$

【０１００】

αは、機器Ａ、機器Ｂ間のメッセージ送受信時の遅延時間と揺らぎを考慮した値である。

【０１０１】

ＭＡＸ（Ｔａ＿ｌｏｎｇ２、Ｔｂ＿ｌｏｎｇ２）は、Ｔａ＿ｌｏｎｇ２とＴｂ＿ｌｏｎｇ２のうち大きい方の値である。このようにすれば、Ｔａ＿ｌｏｎｇ２、または、Ｔｂ＿ｌｏｎｇ２を長めに取ることが可能になる。

【０１０２】

ステップ７０１は、ステップ７０２、ステップ７０３、ステップ７０４、ステップ７０５に必要な時間ｔ＿ＡＢを見込んで早めに開始しなければならない。古い鍵を生成した直後に、ステップ７０１～ステップ７０５を実行して、次のｔ＿ｓｔａｒｔを算定しておけばよい。また、時刻（ｔ＿ｓｔａｒｔ－ｔ＿ＡＢ）が来たら、ステップ７０１～ステップ７０５を再度実行して、その時の最新の所要時間を問い合わせて、ｔ＿ｓｔａｒｔを算定し直すようにしてもよい。

【０１０３】

処理時間の推定方法としては、古い鍵の交換時に行なった各ステップの処理時間、またはＣＰＵ負荷値を、図３のデータベース部３０３に機器Ａ、機器Ｂでそれぞれ記憶しておき、その数値を元にしてもよい。また、ステップ７０１の問い合わせ時における機器Ａ、機器Ｂの各ＣＰＵ負荷値と、上記処理時間、またはＣＰＵ負荷値とから推定してもよい。

【０１０４】

ステップ７０４のあとに機器Ｂにおいて、Ｔａ＿ｌｏｎｇ１＋α以上、ステップ７１０のあとに機器Ａにおいて、Ｔｂ＿ｌｏｎｇ１＋α以上、ステップ７１４のあとに機器Ｂにおいて、Ｔａ＿ｌｏｎｇ２＋α以上のタイムアウト時間を設定することにより、第１の課題も解決できる。

【０１０５】

図７では、機器Ａから所要時間の問い合わせを行ない、ｔ＿ｓｔａｒｔの算定を行なうようにしたが、機器Ａと機器Ｂを入れ替えて逆にしてもよい。

【０１０６】

なお、機器Ｂ側に、課題１のような問題がある場合は、ステップ７０４のあとに機器Ｂにおいて、Ｔａ＿ｌｏｎｇ１＋α以上のタイムアウト値を設定してもよい。

【０１０７】

（実施の形態４）

次に本発明の実施の形態４に係る共有鍵交換方法について説明する。実施の形態４は、従来の共有鍵交換方法における３つの課題のうち、第２の課題を解決するための方法であって、図６において説明した、リキー処理の開始時刻を算出して、その時刻までにリキー処理を開始する方法である。本実施の形態では、それぞれの機器が自分の所用時間だけを基にして、リキー開始時刻を推定する。そして、ＣＰＵ処理能力の低い方の機器が、結果的に、鍵更新のイニシエータになる。

【０１０８】

図８は、本発明の共有鍵交換方式におけるリキー処理の手順を示す図である。機器Ａは、ステップ８０１において、機器Ａが行なう公開値Ｘの演算に必要な時間Ｔａ＿ｌｏｎｇ１と共有鍵Ｋの演算に必要な時間Ｔａ＿ｌｏｎｇ２の推定を行ない、機器Ａ自身のリキー開始時刻を次のように推定する。

【０１０９】

【数９】

t_startA＜t_endA－（Ta_long1＋Ta_long2）

＊２－α

【０１１０】

ここで、t__endAは、機器Aが使用中の共有鍵Kの寿命満了時刻、t__startAは、機器Aのリキー開始時刻である。また、αは、機器A、機器B間のメッセージ送受信時の遅延時間と揺らぎを考慮した値である。

【０１１１】

機器Bは、ステップ８０２において、Tb__long1とTb__long2の推定を行ない、機器B自身のリキー開始時刻を次のように推定する。

【０１１２】

【数１０】

t_startB＜t_endB－（Tb_long1＋Tb_long2）

＊２－α

【０１１３】

ここで、t__endBは、機器Bが使用中の共有鍵Kの寿命満了時刻、t__startBは、機器Bのリキー開始時刻である。また、αは、機器A、機器B間のメッセージ送受信時の遅延時間と揺らぎを考慮した値である。

【０１１４】

Ta__long1、Ta__long2、Tb__long1、Tb__long2は、それぞれ、演算の正味時間に、その前後に必要な付随する処理時間や揺らぎも含んだ値である。また、一般的に、t__endAとt__endBは同じ値に設定されている。

【０１１５】

機器Bが機器Aよりも、ＣＰＵ処理能力が低い場合、t__startAよりt__startBの方が小さい値となり、t__startBが時間的に先になる。

【０１１６】

機器Bは、時刻t__startBが来ると、リキー処理を開始する。すなわち、ステップ８０３において、変数gとnを機器Aに通知し、リキー開始を宣言、

通知する。機器Ａは、ステップ８０４において、変数ｇとｎを受取ったことを応答、通知し、リキー開始を了解する。なお、新たな変数ｇとｎを使わず、前の数値を使う場合は、リキー開始の通知とその了解の応答を行なうだけで良い。

【０１１７】

機器Ｂは、ステップ８０５において、秘密値ｂを生成し、ステップ８０６において、公開値Ｙを演算する。ステップ８０５、ステップ８０６の処理に、Ｔｂ＿ｌｏｎｇ１かかる。機器Ｂは、ステップ８０７において、公開値Ｙを機器Ａに通知する。機器Ａは、自身のリキー開始時刻ｔ＿ｓｔａｒｔＡに関係なく、ステップ８０８において、秘密値ａを生成し、ステップ８０９において、公開値Ｘを演算する。これらの処理に、Ｔａ＿ｌｏｎｇ１の時間がかかる。機器Ａは、つぎに、ステップ８１０において、共有鍵Ｋを演算して生成する。この処理に、Ｔａ＿ｌｏｎｇ２かかる。機器Ａは、ステップ８１１において、公開値Ｘを機器Ｂに通知する。機器Ｂは、ステップ８１２において、共有鍵Ｋを生成する。ステップ８１２の処理に、Ｔｂ＿ｌｏｎｇ２かかる。

【０１１８】

機器Ｂは、ｔ＿ｓｔａｒｔＢを（数１０）とした。すなわち、機器Ｂは、機器Ａの行なうステップ８０５、ステップ８０６、ステップ８０７、ステップ８１２の処理についても、機器Ｂ自身の処理時間がかかると仮定したが、機器ＡのＣＰＵ処理能力は、機器Ｂのそれよりも高いので、ステップ８１２の処理は、ｔ＿ｅｎｄＡ、ｔ＿ｅｎｄＢまでに完了することになる。すなわち、古い鍵の寿命満了以前に、新しい鍵が誕生した。以降は、この新しい鍵が使用される。

【０１１９】

機器Ａ、機器ＢのうちＣＰＵ処理能力が低い方が、先にリキー処理の開始を宣言、通知するので、お互い相手の機器の公開値、共有鍵Ｋの演算所要時間を知る必要が無い。

【０１２０】

ＣＰＵ処理能力の高い方の機器Ａは、ＣＰＵ処理能力の低い方の機器Ｂにしたがって鍵交換処理を開始し処理を進めるので、機器Ａ自身のリキー開始時刻ｔ＿ｓｔａｒｔＡは、機器Ａの各処理ステップの処理時刻とは関係なくなる。

【０１２１】

ステップ８０１、ステップ８０２は、早めに開始しなければならない。古い鍵を生成した直後に、ステップ８０１、ステップ８０２を実行して、次のｔ＿ｓｔａｒｔＡ、ｔ＿ｓｔａｒｔＢを算定しておけばよい。

【０１２２】

処理時間の推定方法としては、古い鍵の交換時に行なった各ステップの処理時間、またはＣＰＵ負荷値を、図３のデータベース部３０３に機器Ａ、機器Ｂでそれぞれ記憶しておき、その数値を元にしてもよい。また、ステップ８０１、ステップ８０２の推測時における機器Ａ、機器Ｂの各ＣＰＵ負荷値と、上記処理時間、またはＣＰＵ負荷値とから推定してもよい。

【０１２３】

なお、第１の課題を解決するには、機器ＢからＴｂ＿ｌｏｎｇ１＋αの時間を入手して、ステップ８０４のあとに機器Ａにおいて、Ｔｂ＿ｌｏｎｇ１＋α以上のタイムアウト時間を設定すればよい。さらに、機器ＢからＴｂ＿ｌｏｎｇ２＋αの時間を入手して、ステップ８１１のあとに機器Ａにおいて、Ｔｂ＿ｌｏｎｇ２＋α以上のタイムアウト時間を設定してもよい。

【０１２４】

（実施の形態５）

次に本発明の実施の形態５に係る共有鍵交換方法について説明する。実施の形態５は、従来の共有鍵交換方法における３つの課題のうち、第２の課題を解決するための方法であって、図６において説明した、リキー処理の開始時刻を算出し、その時刻までにリキー処理を開始する方法である。本実施の形態では、それぞれの機器が自分の所用時間だけを基にして、リキー開始時刻を推定する。そして、ＣＰＵ処理能力の低い方の機器が、結果的に、鍵更新のイニシエータになる。

【０１２５】

図９は、本発明の共有鍵交換方式におけるリキー処理の手順を示す図である。機器Ａは、ステップ９０１において、機器Ａが行なう公開値Ｘの演算に必要な時間Ｔａ＿ｌｏｎｇ１と共有鍵Ｋの演算に必要な時間Ｔａ＿ｌｏｎｇ２の推定を行ない、機器Ａ自身のリキー開始時刻を次のように推定する。

【０１２６】

【数１１】

$$t\_startA < t\_endA - (Ta\_long1 + Ta\_long2) - \alpha$$

【０１２７】

ここで、ｔ＿ｅｎｄＡは、機器Ａが使用中の共有鍵Ｋの寿命満了時刻、ｔ＿ｓｔａｒｔＡは、機器Ａのリキー開始時刻である。また、αは、機器Ａ、機器Ｂ間のメッセージ送受信時の遅延時間と揺らぎを考慮した値である。

【０１２８】

機器Ｂは、ステップ９０２において、Ｔｂ＿ｌｏｎｇ１とＴｂ＿ｌｏｎｇ２の推定を行ない、機器Ｂ自身のリキー開始時刻を次のように推定する。

【０１２９】

【数１２】

$$t\_startB < t\_endB - (Tb\_long1 + Tb\_long2) - \alpha$$

【０１３０】

ここで、ｔ＿ｅｎｄＢは、機器Ｂが使用中の共有鍵Ｋの寿命満了時刻、ｔ＿ｓｔａｒｔＢは、機器Ｂのリキー開始時刻である。また、αは、機器Ａ、機器Ｂ間のメッセージ送受信時の遅延時間と揺らぎを考慮した値である。

【０１３１】

Ｔａ＿ｌｏｎｇ１、Ｔａ＿ｌｏｎｇ２、Ｔｂ＿ｌｏｎｇ１、Ｔｂ＿ｌｏｎｇ２は、それぞれ、演算の正味時間に、その前後に必要な付随する処理時間や揺らぎも含んだ値である。また、一般的に、ｔ＿ｅｎｄＡとｔ＿ｅｎｄＢは同じ値に設定されている。

【０１３２】

機器Ｂが機器Ａよりも、ＣＰＵ処理能力が低い場合、ｔ＿ｓｔａｒｔＡよりｔ＿ｓｔａｒｔＢの方が小さい値となり、ｔ＿ｓｔａｒｔＢが時間的に先になる。

【０１３３】

機器Ｂは、時刻ｔ＿ｓｔａｒｔＢが来ると、リキー処理を開始する。すなわち、ステップ９０３において、変数ｇとｎを機器Ａに通知し、リキー開始を宣言、通知する。機器Ａは、ステップ９０４において、変数ｇとｎを受取ったことを応

答、通知し、リキー開始を了解する。なお、新たな変数ｇとｎを使わず、前の数値を使う場合は、リキー開始の宣言、通知とその了解の応答、通知を行なうだけで良い。

【０１３４】

機器Ｂは、ステップ９０５において秘密値ｂを生成し、ステップ９０６において公開値Ｙを演算する。これらの処理に、Ｔｂ＿ｌｏｎｇｌの時間がかかる。機器Ｂは、ステップ９１０において公開値Ｙを機器Ａに通知する。機器Ａは、ステップ９１２において、共有鍵Ｋを演算して生成する。

【０１３５】

機器Ａは、ステップ９０７において秘密値ａを生成し、ステップ９０８において公開値Ｘを演算し、ステップ９０９において公開値Ｘを機器Ｂに通知する。ステップ９０７、ステップ９０８の処理に、Ｔａ＿ｌｏｎｇｌかかる。機器Ｂは、ステップ９１１において、共有鍵Ｋを演算して生成する。この処理に、Ｔｂ＿ｌｏｎｇ２かかる。

【０１３６】

機器Ｂより機器ＡのＣＰＵ処理能力が高い場合は、ステップ９０９がステップ９１０よりも先に生起する場合が多くなる。

【０１３７】

機器Ｂは、ｔ＿ｓｔａｒｔＢを（数１２）とした。すなわち、機器Ｂは、機器Ａの行なうステップ９０７、ステップ９０８、ステップ９０９、ステップ９１２の処理についても、機器Ｂ自身の処理時間がかかると仮定したが、機器ＡのＣＰＵ処理能力は、機器Ｂのそれよりも高いので、ステップ９１１、ステップ９１２の処理は、ｔ＿ｅｎｄＡ、ｔ＿ｅｎｄＢまでに完了することになる。すなわち、古い鍵の寿命満了以前に、新しい鍵が誕生した。以降は、この新しい鍵が使用される。

【０１３８】

機器Ａ、機器ＢのうちＣＰＵ処理能力が低い方が、先にリキー処理の開始を宣言するので、お互い相手の機器の公開値、共有鍵Ｋの演算所要時間を知る必要が無い。

【０１３９】

ＣＰＵ処理能力の高い方の機器Ａは、ＣＰＵ処理能力の低い方の機器Ｂにしたがって鍵交換処理を開始し処理を進めるので、機器Ａ自身のリキー開始時刻ｔ＿ｓｔａｒｔＡは、機器Ａの各処理ステップの時刻とは関係なくなる。

【０１４０】

ステップ９０１、ステップ９０２は、早めに開始しなければならない。古い鍵を生成した直後に、ステップ９０１、ステップ９０２を実行して、次のｔ＿ｓｔａｒｔＡ、ｔ＿ｓｔａｒｔＢを算定しておけばよい。

【０１４１】

処理時間の推定方法としては、古い鍵の交換時に行なった各ステップの処理時間、またはＣＰＵ負荷値を、図３のデータベース部３０３に機器Ａ、機器Ｂでそれぞれ記憶しておき、その数値を元にしてもよい。また、ステップ９０１、ステップ９０２の推測時における機器Ａ、機器Ｂの各ＣＰＵ負荷値と、上記処理時間、またはＣＰＵ負荷値とから推定してもよい。

【０１４２】

なお、第１の課題を解決するには、機器ＢからＴｂ＿ｌｏｎｇ１＋αの時間を入手して、ステップ９０４のあとに機器Ａにおいて、Ｔｂ＿ｌｏｎｇ１＋α以上のタイムアウト時間を設定してもよい。さらに、機器ＢからＴｂ＿ｌｏｎｇ２＋αの時間を入手して、ステップ９０９のあとに機器Ａにおいて、Ｔｂ＿ｌｏｎｇ２＋α以上のタイムアウト時間を設定してもよい。

【０１４３】

（実施の形態６）

次に本発明の実施の形態６に係る共有鍵交換方法について説明する。実施の形態６は、従来の共有鍵交換方法における３つの課題のうち、第３の課題を解決するための方法である。

【０１４４】

まず、実施の形態６の概要について説明する。ネット家電のように性能の低いＣＰＵを搭載している機器で、Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎのような高負荷の処理を実行すると、ＣＰＵが占有されてしまい、長い処理期間中、他のアプリケ

ーションが正しく動作しなくなる可能性がある。

【０１４５】

一方、共有鍵交換は、常に最短時間で完了させる必要はない。具体的には、パケット通信を行う端末の要請により、鍵を新規に交換する場合には、鍵交換の遅延時間が端末のアプリケーションでの遅延としてユーザに見えてしまうので、できるだけ早く鍵交換を完了させる必要がある。しかしながら、一旦鍵を交換してしまった後、定期的に鍵を更新する場合は、ユーザからのパケットを古い鍵で暗号化処理するのと並行に、新しい鍵の交換を行うので、鍵交換自体の処理時間は長くてもよい。すなわち、古い鍵の寿命が切れる前に更新用の鍵交換が完了すれば良い。

【０１４６】

以上のことから、本発明の実施の形態６では、鍵交換の処理時間を最短で完了させるべきか否かを判定し、最短の処理が必要とされない状況の場合は、高負荷の処理を小単位に分けて実行し、時間的に負荷分散する。この処理により、鍵交換に伴う処理がＣＰＵを占有せず、他のアプリケーションの動作を妨げないようにできる。また、対向の機器への応答として公開値を送信するまでの遅延時間が長くなるが、鍵の更新のように鍵交換の処理遅延の長さがユーザのパケット送受信に影響を与えない状況なので、問題にならない。

【０１４７】

以下、図１０を使って、本発明の実施の形態６に係る共有鍵交換方法の概要を説明する。図１０において、機器Ａと機器Ｂは、暗号・認証処理と共有鍵交換を終端する機器の組とする。具体的には、図２中の機器間で共有鍵交換の処理を終端する３つの組み合わせ、すなわち（１）ＧＷ型機器Ａ２０１とＧＷ型機器Ｂ２０２、（２）ＧＷ型機器Ａ２０１とホスト型機器Ｂ２１１、（３）ホスト型機器Ａ２１０とホスト型機器Ｂ２１１のいずれかの組み合わせに対応するものとする。また、図１０中の機器Ｂは、機器Ａに比べて低い性能のＣＰＵを搭載した機器とする。また、図１０の機器Ａまたは機器Ｂに関する処理シーケンスは、図３における共有鍵交換部３０１で行う処理に対応するものとする。

【０１４８】

まず、機器Ｂでは事前に、Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎ交換の公開値Ｙと共有鍵Ｋの演算に伴う全処理を、複数の均等な処理単位（以下、小処理単位と呼ぶ）に分けておく。さらに、単位時間の間に小処理単位を１単位分実行したときのＣＰＵ使用率：Ｕｃｐｕを実測した結果を記録しておくものとする。ここで、機器Ｂが行うべき全演算に対する小処理単位の数を、Ｔｏｔａｌとする。

【０１４９】

また、機器Ａと機器Ｂの間では、既に共有鍵Ｋ＿ｏｌｄを共有しており、その鍵の寿命が未来の時刻：ｔ＿ｅｎｄに切れる（使用不可の状態になる）ものとする。

【０１５０】

機器Ａは、共有鍵Ｋ＿ｏｌｄの鍵交換中または交換後に、機器Ｂに対して、鍵更新時の所要時間を問い合わせる（ステップ１００１）。機器Ｂでは、その時点の他のアプリケーションの平均的なＣＰＵ使用率を測定し、他のアプリケーションの処理を維持したまま、Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎの演算にどれだけのＣＰＵ使用率を使えるか、単位時間に実行できる小処理単位の数はいくらかを推定する。具体的には、以下の式により、機器Ｂから機器Ａに応答を返すまでの遅延時間を求める（ステップ１００２）。

【０１５１】

【数１３】

Ｄｃｐｕ：残りＣＰＵ使用率

＝１００－他のアプリケーションのＣＰＵ使用率平均

【０１５２】

【数１４】

単位時間の間に実行可能なＤｉｆｆｉｅ－Ｈｅｌｌｍａｎの処理単位数

＝Ｄｃｐｕ／Ｕｃｐｕ

【０１５３】

【数１５】

遅延時間：Ｔｂ_ｌｏｎｇ

＝Ｔｏｔａｌ×Ｕｃｐｕ／Ｄｃｐｕ＋α

ただし、α：固定値

【０１５４】

機器Ｂは、推定した遅延時間：Ｔｂ__ｌｏｎｇを、機器Ａに送信する（ステップ１００３）。

【０１５５】

一方、機器Ａ側でも自分のＤｉｆｆｉｅ－Ｈｅｌｌｍａｎに伴う遅延時間：Ｔａ__ｌｏｎｇを推定しておき、共有鍵Ｋ__ｏｌｄを更新するために鍵交換を開始すべき時刻：ｔ__ｓｔａｒｔを求める（ステップ１００４）。

【０１５６】

【数１６】

ｔ_ｓｔａｒｔ＝ｔ_ｅｎｄ－（Ｔａ_ｌｏｎｇ＋Ｔｂ_ｌｏｎｇ）

【０１５７】

機器Ａでは、共有鍵Ｋ__ｏｌｄの更新開始タイミングとして、時刻：ｔ__ｓｔａｒｔが到来していないかチェックし、ｔ__ｓｔａｒｔの時刻（ステップ１００５）になると、以下の手順で鍵交換を実行する。

【０１５８】

機器Ａは、Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎの公開値Ｘを計算し、機器Ｂに送信する（ステップ１００６～１００８）。機器Ｂは公開値Ｘを受信すると、その要求が既に存在する鍵の更新用なのか、新規に鍵を作成するための要求かを判定する（ステップ１００９）。具体的には、鍵の情報を記憶させているデータベース（図３中のデータベース部に対応）を検索し、対応する鍵が存在する場合は「鍵の更新」、存在しない場合は「新規作成」と判定する。機器Ｂは、判定結果に応じて以後のＤｉｆｆｉｅ－Ｈｅｌｌｍａｎの演算の実行方法を変える。すなわち、鍵の「新規作成」と判定された場合には、機器Ｂとしての最短の時間で実行し、「鍵の更新」と判定された場合には、機器Ａに通知したＴｂ__ｌｏｎｇの時間をかけてゆっくり処理する。図１０のシーケンスでは、Ｋ__ｏｌｄが既に存在す

るので、「鍵の更新」と判定される。機器Ｂは単位時間あたりに、Ｄｃｐｕ／Ｕｃｐｕ個の小処理単位を実行し、Ｔｂ＿ｌｏｎｇの時間をかけて公開値Ｙを算出し、機器Ａに送信する（ステップ１０１０～１０１３）。

【０１５９】

一方、機器Ａも、少なくとも機器Ｂから通知されたＴｂ＿ｌｏｎｇの時間だけ、機器Ｂからの応答を待ち、公開値Ｙを受信した後、新しい共有鍵Ｋ＿ｎｅｗを算出する（ステップ１０１４）。

【０１６０】

以上のように、本発明の実施の形態６では、鍵の更新のような鍵交換の処理遅延時間が問題にならないケースに限定して、鍵交換に伴う高負荷の演算を時間的に負荷分散する。これにより、低性能のＣＰＵを搭載した機器でも、鍵交換の処理が長期間にわたってＣＰＵを占有することがなくなり、同じＣＰＵ上で動作する他のアプリケーションも正しく動作できるようになる。さらに、高負荷の演算を時間的に負荷分散する際にも、鍵の寿命（使用不可になるタイミング）を考慮することで、鍵交換の処理遅延時間が通常より長くなっても、暗号・認証処理を問題なく継続できる。なお、一例として、機器Ｂにおいて、Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎの演算に必要なＣＰＵの処理量がＩｍ＝２００（ＭＩ＝メガインストラクション）であり、複数の均等な処理単位、すなわち、小処理単位として２（ＭＩ）に分ける場合、Ｔｏｔａｌの値は、１００（処理単位）である。機器ＢのＣＰＵの処理能力が１００（ＭＩＰＳ＝メガインストラクション／秒）の場合、小処理単位の２（ＭＩ）を１単位分だけ、単位時間の間に実行するときのＣＰＵ使用率Ｕｃｐｕは、２％である。機器ＢのＣＰＵの処理能力を他のアプリケーションに５０ＭＩＰＳだけ使用する場合、残りＣＰＵ使用率Ｄｃｐｕは、５０％である。従って、単位時間に実行可能なＤｉｆｆｉｅ－Ｈｅｌｌｍａｎの処理単位数Ｄｃｐｕ／Ｕｃｐｕは、２５（処理単位／秒）である。よって、Ｔｏｔａｌ×Ｕｃｐｕ／Ｄｃｐｕは、１００／２５＝４秒になる。２０％の揺らぎなどの余裕を見込めば、遅延時間は、４．８秒とすればよい。

【０１６１】

（実施の形態７）

図１０の実施の形態６において、処理の不可分散を良く行える方法を次に説明する。このために、実施の形態７では、鍵の交換が終了したら、機器Ａ、Ｂは、可能なかぎり速やかに共有鍵の更新処理に入る。まず、機器Ｂは、ステップ１００２においてＣＰＵの使用量Ｉｍの推定を行なう。ＣＰＵの使用量Ｉｍは、例えば、機器Ｂが実施した鍵の作成処理の実行命令数（単位：ＭＩ：メガインストラクション）で表すことができる。前回のＣＰＵ使用量Ｉｍを記憶しておいてその値を使用してもよい。機器Ａは、ステップ１００４において、自身の秘密値ａの生成、公開値Ｘの演算、共有鍵Ｋの演算に必要な時間の合計値（Ｔａ＿ｌｏｎｇ１＋Ｔａ＿ｌｏｎｇ２）を（鍵の寿命完了時刻－現在時刻）から差し引いた機器Ｂ用割り当て時間ＴＷｂを求め、ステップ１００８において、公開値Ｘと共に、機器Ｂに通知する。機器Ｂは、ステップ１００９において、単位時間当たりのＣＰＵ使用率（Ｉｍ／ＴＷｂ＋β）をＣＰＵに割り当てて、ステップ１０１０の秘密値ｂの生成、ステップ１０１１の公開値Ｙの演算、ステップ１０１２の共有鍵Ｋの演算を行なう。βは、処理の揺らぎや前後に必要な処理を行なうための余裕分である。一般的に、ＴＷｂを大きくすれば、単位時間当たりのＣＰＵ使用率（Ｉｍ／ＴＷｂ＋β）を、１００％より十分小さい値にすることができる。そうすれば、残ったＣＰＵ使用率の部分を、鍵交換処理以外のアプリケーションに振り向ける余裕ができる。Ｔｂ＿ｌｏｎｇの代りに、機器Ｂ用割り当て時間ＴＷｂ＋αが、タイムアウト待ち時間として設定される。図１０において、ステップ１００１は不要になる。

【０１６２】

このようにすれば、機器Ｂにおいて、鍵交換処理に割く単位時間当たりのＣＰＵ処理量を常に低くできるので、他の処理に回せるＣＰＵ処理量を多めに確保しておける。また、上記単位時間当たりのＣＰＵ使用率（Ｉｍ／ＴＷｂ＋β）を割り当ての最小量とし、他の処理の生起が少なくてＣＰＵ処理能力に余裕がある時間帯には、より大きいＣＰＵ使用率を割り当てて、早めに鍵交換処理を進めるようにしてもよい。

【０１６３】

なお、本実施の形態と同様の考え方を、図８の（実施の形態４）の演算におい

ても、適用できる。

【０１６４】

（実施の形態８）

図９の実施の形態５において、処理の負荷分散をよく行なえる方法を次に説明する。このために、実施の形態８では、鍵の交換が終了したら、機器Ａ、Ｂは、できる限り速やかに鍵の更新処理に入る。まず、機器Ｂは、ステップ９０２においてＣＰＵの使用量Ｉｍの推定を行なう。前回のＣＰＵ使用量Ｉｍを記憶しておいてその値を使用してもよい。機器Ｂは、余裕時間ＴＹｂ＝（鍵の寿命満了時刻－現在時刻）を算出し、単位時間当たりのＣＰＵ使用率＝（Ｉｍ／ＴＹｂ＋β）をＣＰＵに割り当てて、ステップ９０５の秘密値ｂの生成、ステップ９０６の公開値Ｙの演算、ステップ９１１の共有鍵Ｋの演算を行なう。βは、処理の揺らぎや各演算の前後に必要な処理を行なうための余裕分である。一般的に、ＴＹｂを大きくすれば、単位時間当たりのＣＰＵ使用率（Ｉｍ／ＴＹｂ＋β）を、１００％より十分小さい値にすることができる。そうすれば、残ったＣＰＵ使用率の部分を、鍵交換処理以外のアプリケーションに振り向ける余裕ができる。

【０１６５】

一方、機器Ａは、ステップ９０１において自身のＣＰＵ使用量の推定を行なう。前回のＣＰＵ使用量を記憶しておいてその値を使用してもよい。前回のＣＰＵ使用量は、機器Ａにおいても、Ｉｍである。機器Ａは、余裕時間ＴＹａ＝（鍵の寿命満了時刻－現在時刻）を算出し、単位時間当たりのＣＰＵ使用率＝（Ｉｍ／ＴＹａ＋β）をＣＰＵに割り当てて、ステップ９０７の秘密値ａの生成、ステップ９０８の公開値Ｘの演算、ステップ９１２の共有鍵Ｋの演算を行なう。なお、機器ＡにおけるＣＰＵ使用量が、機器Ｂの場合とは異なる場合には、機器ＡにおけるＣＰＵ使用量を用いればよい。

【０１６６】

なお、ステップ９０３、ステップ９０４のリキー開始の宣言は、なくともよい。また、機器Ａ、Ｂの演算進行速度が、ほぼ同じになるので、ステップ９０９の公開値Ｘの通知、ステップ９１０の公開値Ｙの通知は、ほぼ同じ時刻に行われることになり、機器ＢのＣＰＵ処理能力が低くとも、ステップ９１１の処理の完了

が、ｔ＿ｅｎｄＢに間に合わなくなることはない。

【０１６７】

このようにすれば、機器Ｂにおいて、鍵交換処理に割く単位時間当たりのＣＰＵ処理量を常に低くできるので、他の処理に回せるＣＰＵ処理量を多めに確保しておける。また、上記単位時間当たりのＣＰＵ使用率（Ｉｍ／ＴＹｂ＋β）を割り当ての最小量とし、他の処理の生起が少なくてＣＰＵ処理能力に余裕がある時間帯には、より大きいＣＰＵ使用率を割り当てて、早めに鍵交換処理を進めるようにしてもよい。

【０１６８】

（実施の形態９）

上記、各実施の形態において、第１の通信機器、第２の通信機器が、共有鍵Ｋの演算を終了したら、相手の機器に対して、共有鍵Ｋの生成完了の通知を行なうようにしてもよい。この場合、この通知を受信するまで、タイムアウトの検出を停止するようにしてもよい。そうすれば、第１の課題を解決できる。

【０１６９】

（実施の形態１０）

先に、鍵の寿命を設定できることを説明した。この設定は、一般的に、図１３の各種パラメータの折衝ステップ１２０２において行われる。この寿命を、（Ｔａ＿ｌｏｎｇ１＋Ｔａ＿ｌｏｎｇ２）と（Ｔｂ＿ｌｏｎｇ１＋Ｔｂ＿ｌｏｎｇ２）を基準にして、十分余裕をもった値に設定することにより、鍵の寿命に到達する前に、リキーを完了できるようにしてもよい。鍵交換の完了時間が、（Ｔａ＿ｌｏｎｇ１＋Ｔａ＿ｌｏｎｇ２）と（Ｔｂ＿ｌｏｎｇ１＋Ｔｂ＿ｌｏｎｇ２）の和によって決まる、図７の（実施の形態３）、図８の（実施の形態４）の場合は、（Ｔａ＿ｌｏｎｇ１＋Ｔａ＿ｌｏｎｇ２）＋（Ｔｂ＿ｌｏｎｇ１＋Ｔｂ＿ｌｏｎｇ２）により、図９の（実施の形態５）の場合は、（Ｔａ＿ｌｏｎｇ１＋Ｔａ＿ｌｏｎｇ２）と（Ｔｂ＿ｌｏｎｇ１＋Ｔｂ＿ｌｏｎｇ２）のうち、大きい方の値を基準にすればよい。例えば、機器ＢのＣＰＵ処理能力が大きくなく、（Ｔｂ＿ｌｏｎｇ１＋Ｔｂ＿ｌｏｎｇ２）が長い場合、ＣＰＵ処理能力が低いほど、鍵の寿命を長くすればよい。

【０１７０】

上記基準の時間の数倍、あるいは１０倍程度の寿命を設定すれば、ＣＰＵ処理能力が低くとも、鍵交換処理以外の処理に振り向けるＣＰＵ処理能力の余裕が生れる。

【０１７１】

なお、（Ｔａ＿ｌｏｎｇ１＋Ｔａ＿ｌｏｎｇ２）と（Ｔｂ＿ｌｏｎｇ１＋Ｔｂ＿ｌｏｎｇ２）としては、最初の鍵交換処理の際にかかった処理時間を記憶しておき、その値を使用してもよい。

【０１７２】

図１の実施の形態１においては、機器Ｂは、ステップ１０７において、応答の予定時間遅れを通知し、機器Ａは、その時間だけ、タイムアウトを遅らせるようにした。また、図５の実施の形態２においては、状況確認・再送に対して公開値の送信が、遅れることを、機器Ｂが機器Ａに通知し、機器Ａがタイムアウトを発生させないようにした。このように、本発明では、応答が遅れることを応答予告として相手に通知することにより、相手側では、種々の対応策を採用することができるようになり、タイムアウトなどによる破綻を防止することを可能にする。

【０１７３】

上記、図５の実施の形態２において、課題１の対策、および、通信のパケットロスが発生する場合の対策として、状況確認・再送を行なったが、他の実施の形態においても、パケットロスの発生によるタイムアウトを避けるために、状況確認・再送を随時行ない、タイムアウト時間を更新するようにしてもよい。

【０１７４】

図８の実施の形態４、図９の実施の形態５において、ｇ、ｎの通知と了解の後に、秘密値ａ、または、ｂの生成を行なうようにしたが、秘密値と公開値の生成を行なっておき、ｇ、ｎ、公開値Ｘ、または、Ｙを、同時に通知するようにしてもよい。

【０１７５】

上記各実施の形態においては、図９の実施の形態５を除いて、最初に、ｇ、ｎを通知したり、パラメータの折衝を開始する側の機器をイニシエータとし、イニ

シエータの方が、秘密値の生成、公開値の演算により得た公開値を、先に相手方に送る例を、主として説明した。反対に、イニシエータでない方の機器が、先に公開値を算出して送る場合でも、本発明を適用することが可能である。

【０１７６】

上記、図７、図８、図９で説明した、鍵の寿命満了までにリキーを行なう各実施の形態において、ｔ＿ｅｎｄＡ、ｔ＿ｅｎｄＢは、ほぼ同じ時刻になるとして説明したが、鍵の寿命は、機器Ａ、機器Ｂにおいて、それぞれの共有鍵の生成完了時点から起算して設定されるように規定されている場合がある。一般的に、機器Ａ、機器Ｂの共有鍵の生成完了時点は、共有鍵の生成手順と、機器Ａ、機器ＢのＣＰＵ処理能力などによって左右され、同時刻となることは希であり、ｔ＿ｅｎｄＡとｔ＿ｅｎｄＢの時刻には、Ｔａ＿ｌｏｎｇ２、あるいは、Ｔｂ＿ｌｏｎｇ２程度の差がでる。

【０１７７】

図７の場合は、ｔ＿ｅｎｄＡは、ｔ＿ｅｎｄＢより（Ｔａ＿ｌｏｎｇ２）だけ遅くなる。（ステップ７１２）の直後にステップ７１４を実行する場合は、ｔ＿ｅｎｄＢとｔ＿ｅｎｄＡの時刻の差、（ｔ＿ｅｎｄＢ－ｔ＿ｅｎｄＡ）は、（Ｔｂ＿ｌｏｎｇ２－Ｔａ＿ｌｏｎｇ２）になる。

【０１７８】

図８の場合は、Ｔ＿ｅｎｄＡは、Ｔ＿ｅｎｄＢより（Ｔｂ＿ｌｏｎｇII）だけ早くなる。ステップ８０９の直後にステップ８１１を実行する場合は、ｔ＿ｅｎｄＢとｔ＿ｅｎｄＡの時刻の差、（ｔ＿ｅｎｄＢ－ｔ＿ｅｎｄＡ）は、（Ｔｂ＿ｌｏｎｇ２－Ｔａ＿ｌｏｎｇ２）になる。

【０１７９】

図９の場合は、ステップ９１１とステップ９１２は、Ｔａ＿ｌｏｎｇ１とＴｂ＿ｌｏｎｇ１の遅い方が完了した後、同時に処理を開始することになるので、ｔ＿ｅｎｄＢとｔ＿ｅｎｄＡの時刻の差、（ｔ＿ｅｎｄＢ－ｔ＿ｅｎｄＡ）は、（Ｔｂ＿ｌｏｎｇ２－Ｔａ＿ｌｏｎｇ２）になる。

【０１８０】

したがって、（数７）～（数１２）におけるｔ＿ｅｎｄＡ、ｔ＿ｅｎｄＢの値

は、上記（ｔ＿ｅｎｄＢ－ｔ＿ｅｎｄＡ）の絶対値に相当する値に所定の余裕を加えた値だけ、時刻の早い数値を採用して計算するのが安全である。一般的に、秘密値、公開値、共有鍵の演算に要する時間の数倍から１０倍以上の時間を鍵の寿命とする場合が多いので、ｔ＿ｅｎｄＡ、ｔ＿ｅｎｄＢの値をＴａ＿ｌｏｎｇ２、あるいは、Ｔｂ＿ｌｏｎｇ２程度早めの値とすることは、十分に可能である。なお、（数７）～（数１２）をそのまま使用するには、鍵の満了時間が、設定した数値よりも上記（ｔ＿ｅｎｄＢ－ｔ＿ｅｎｄＡ）に相当する値程度だけ短いものとして、ｔ＿ｅｎｄＡ、ｔ＿ｅｎｄＢの時刻を設定するようにすればよい。

【０１８１】

なお、鍵の寿命は、１）鍵を生成した時刻から起算する場合、２）鍵を使用して暗号通信を始めた時点から起算する場合、３）暗号通信において一定数のパケットを処理するまで、４）一定のバイト数を通信処理するまで、などの中から選択できるようになっている場合がある。２）鍵を使用して暗号通信を始めた時点から起算する場合、３）暗号通信において一定数のパケットを処理するまで、４）一定のバイト数を通信処理するまで、を選択した場合は、機器Ａと機器Ｂのそれぞれの鍵の寿命満了時刻は、実質的に、同一時刻となる。したがって、上述した、鍵の寿命満了時刻を早めとする配慮は無くともよく、適当な余裕値分だけ早めとすれば十分である。

【０１８２】

ステップ１０７、ステップ４０３で説明した時間Ｔｂ以内の応答を予告する応答予告のメッセージは、応答の遅延時間、応答の送信予定時刻などを送信すればよい。受信した機器Ａは、応答予告の時間または時刻を基に、タイムアウトの時間設定を行なえばよい。

【０１８３】

本発明は、暗号・認証処理を施したデータを送受信する２つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、前記２つの通信機器は、第１の通信機器と第２の通信機器から成り、前記第１の通信機器および前記第２の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものであれば、Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎの

方法やＩＫＥの方法以外の共有鍵交換方法に対しても適用可能である。

【０１８４】

なお、本発明の共有鍵交換方法のプログラムを記録した記録媒体は、プログラムを記録したＲＯＭ、ＲＡＭ、フレキシブルディスク、ＣＤ－ＲＯＭ、ＤＶＤ、メモリカード、ハードディスクなどの記録媒体をいう。また、電話回線、搬送路などの通信媒体も含む概念である。

【０１８５】

【発明の効果】

以上のように、本発明によれば、実施の形態１、実施の形態２、実施の形態６のようにすることにより、対向の機器でのタイムアウト検出により、鍵交換が失敗するという、第１の課題が解決される。また、本発明によれば、実施の形態３、実施の形態４、実施の形態５、実施の形態６、実施の形態７、実施の形態８、実施の形態１０のようにすることにより、鍵の寿命が設定されている場合に、寿命満了までに、次の鍵交換が完了しないという第２の課題が解決される。

【０１８６】

また、本発明によれば、実施の形態６、実施の形態７、実施の形態８、実施の形態１０のようにすることにより、共有鍵交換の処理が、機器上の他のアプリケーションの実行を妨害するという第３の課題が解決される。

【０１８７】

よって、第２の通信機器の搭載するＣＰＵの性能が低い場合にも、負荷の高い鍵交換処理を正しく実行でき、共有鍵交換を成功させることができる。

【図面の簡単な説明】

【図１】

本発明の実施の形態１における共有鍵交換方法の処理手順を示す図

【図２】

本発明の共有鍵交換方法が適用されるネットワークの全体構成を示す図

【図３】

本発明の共有鍵交換方法が適用される機器の機能的構成を示すブロック図

【図４】

本発明の実施の形態１における共有鍵交換方法の別の処理手順を示す図

【図５】

本発明の実施の形態２における共有鍵交換方法の処理手順を示す図

【図６】

本発明の実施の形態３における鍵更新のタイムチャート

【図７】

本発明の実施の形態３における共有鍵交換方法の処理手順を示す図

【図８】

本発明の実施の形態４における共有鍵交換方法の処理手順を示す図

【図９】

本発明の実施の形態５における共有鍵交換方法の処理手順を示す図

【図１０】

本発明の実施の形態６における共有鍵交換方法の処理手順を示す図

【図１１】

従来の共有鍵交換方法であるＤｉｆｆｉｅ－Ｈｅｌｌｍａｎの基本的な処理手順を示す図

【図１２】

ＲＦＣ２４０７～２４０９で開示されているＩＫＥの処理手順の概要を示す図

【図１３】

従来の共有鍵交換方法の問題点を説明するための図

【符号の説明】

２０１　ＧＷ型機器Ａ

２０２　ＧＷ型機器Ｂ

２０３，２０４，２０５，２０６　ＬＡＮ内端末

２０７　公衆網

２０８，２０９　ＬＡＮ

２１０　ホスト型機器Ａ

２１１　ホスト型機器Ｂ

３０１　共有鍵交換部

３０２　データベース管理部

３０３　データベース部

３０４　暗号化／認証処理部

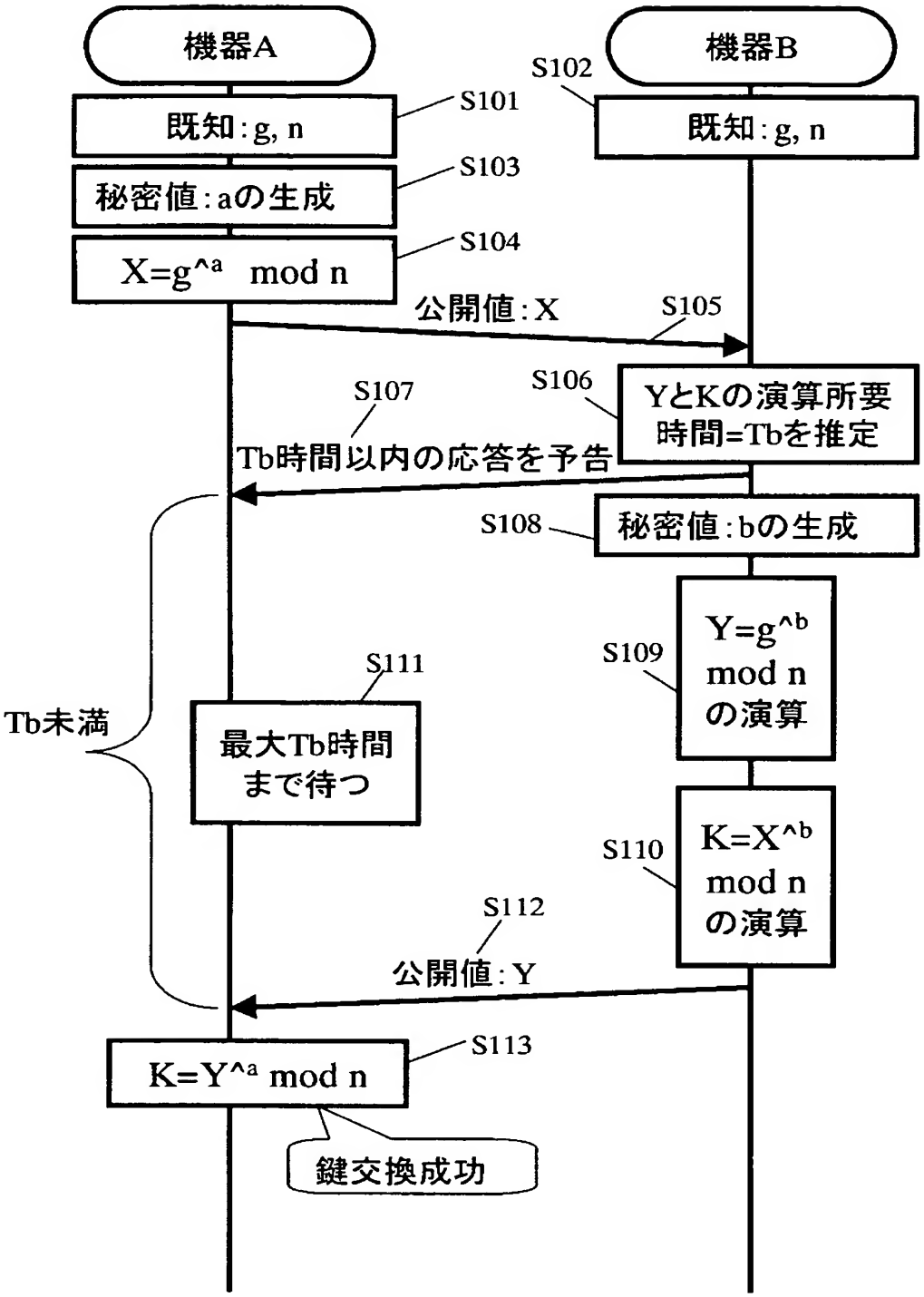
３０５，３０６　通信プロトコル処理部

３０７　ＬＡＮ　Ｉ／Ｆ

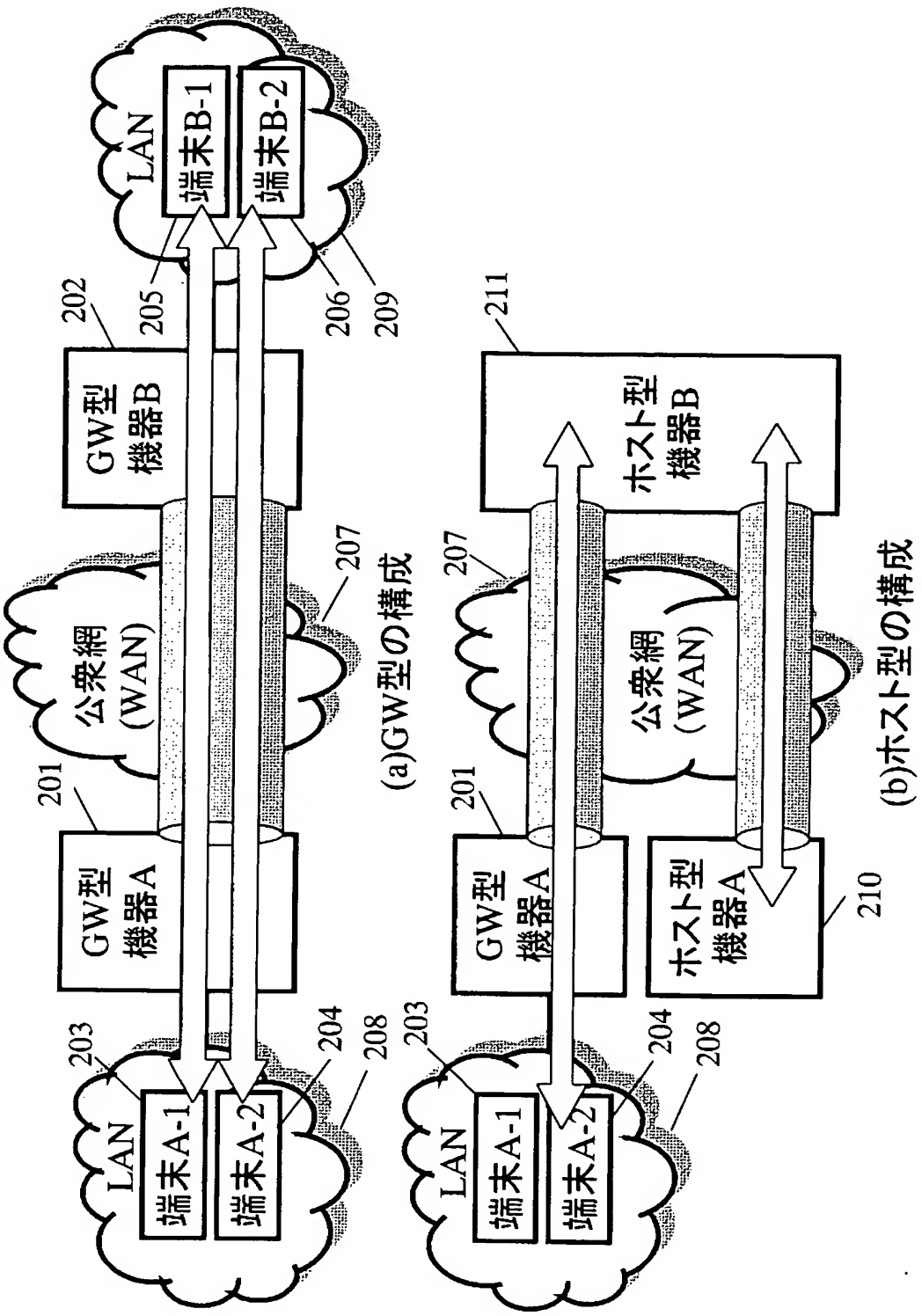
３０８　ＷＡＮ　Ｉ／Ｆ

【書類名】　図面

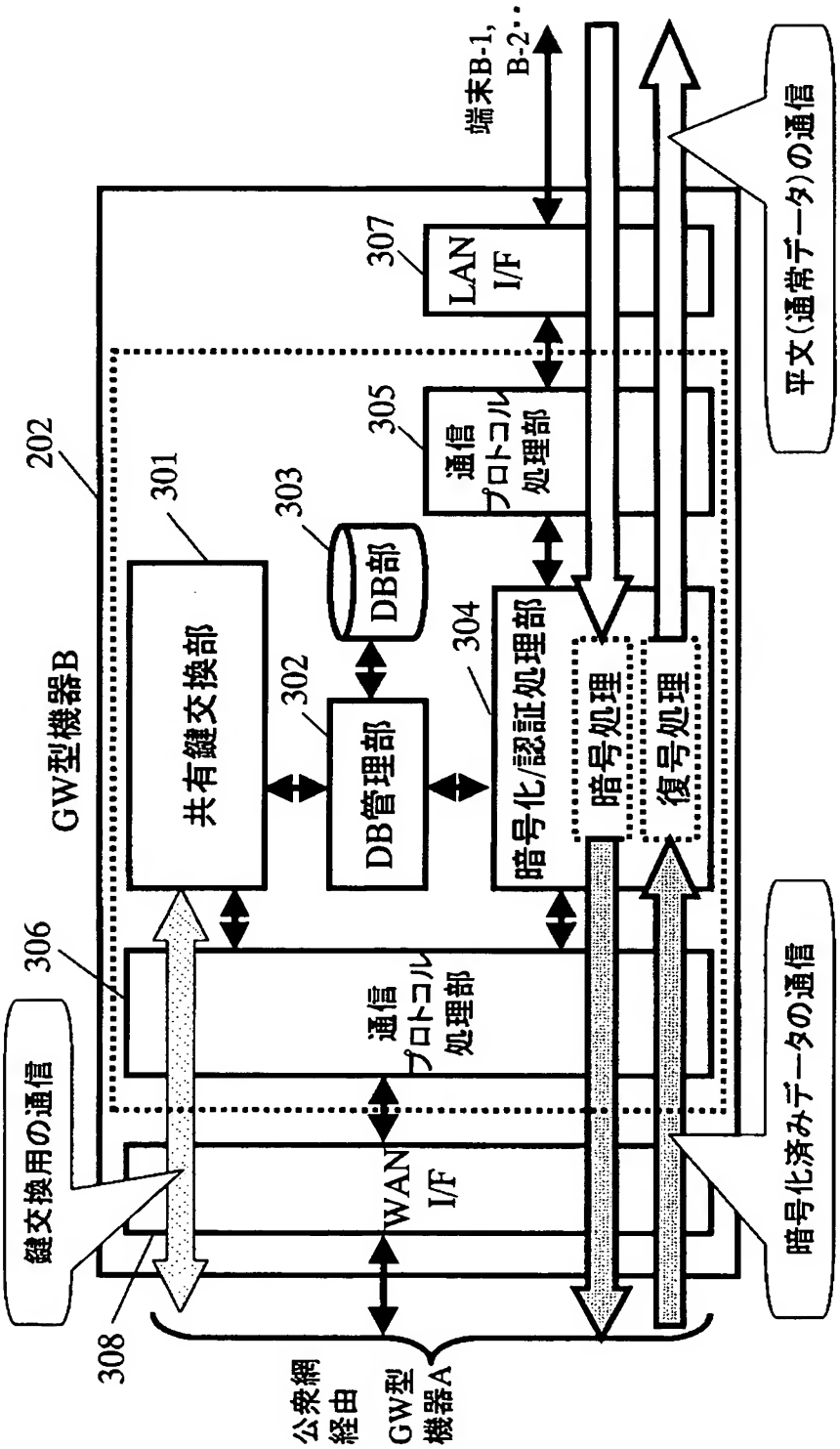
【図１】


機器A
機器B
S101
既知：g, n
S102
既知：g, n
S103
秘密値：aの生成
S104
X=g^a mod n
公開値：X
S105
S106
YとKの演算所要時間=Tbを推定
S107
Tb時間以内の応答を予告
S108
秘密値：bの生成
S109
Y=g^b mod n の演算
S110
K=X^b mod n の演算
S111
最大Tb時間まで待つ
Tb未満
S112
公開値：Y
S113
K=Y^a mod n
鍵交換成功

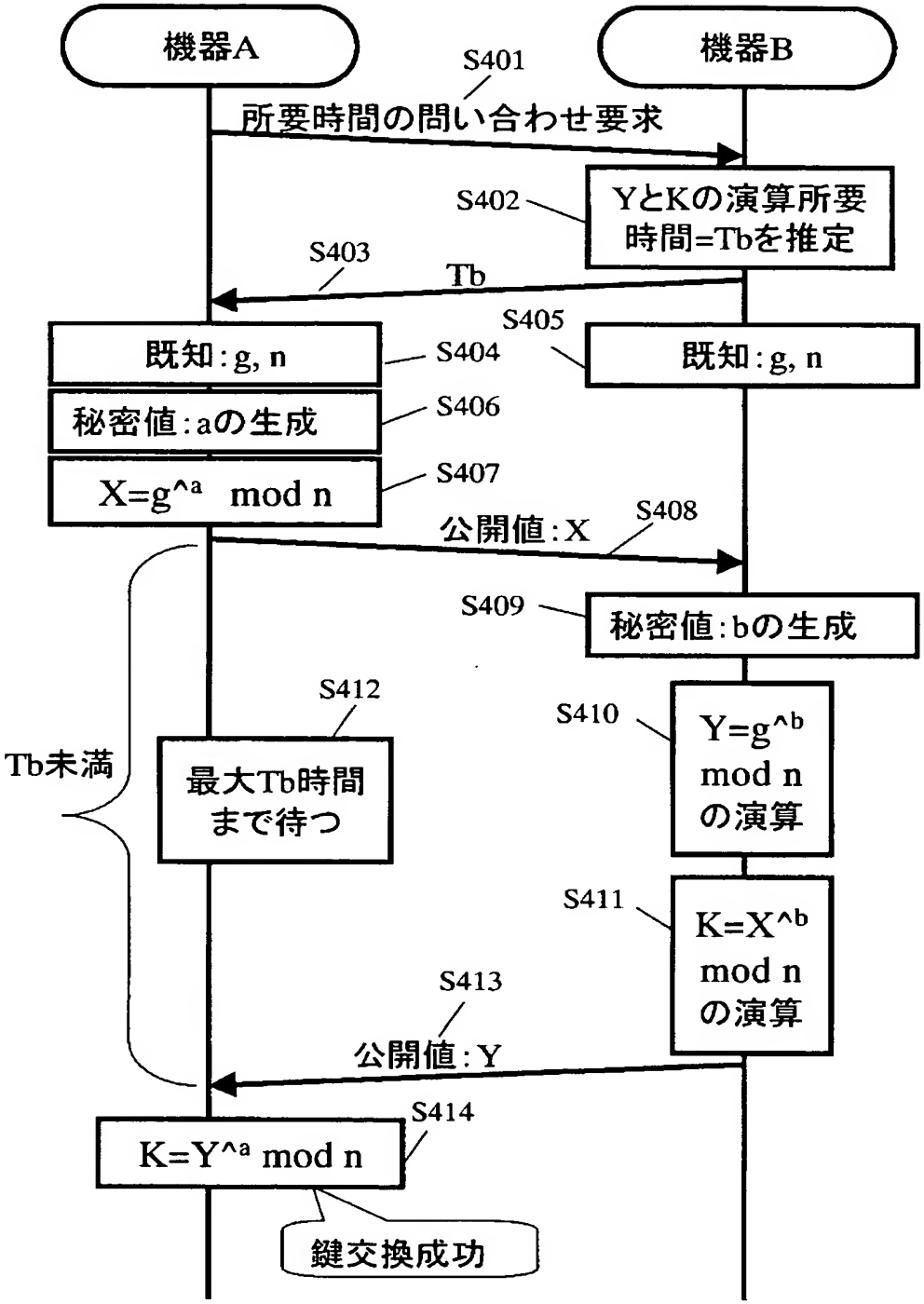
【図２】


LAN
端末A-1
端末A-2
LAN
端末B-1
端末B-2
GW型
機器A
GW型
機器B
公衆網
(WAN)
201
202
203
204
205
206
207
208
209
(a)GW型の構成
ホスト型
機器A
ホスト型
機器B
210
211
(b)ホスト型の構成

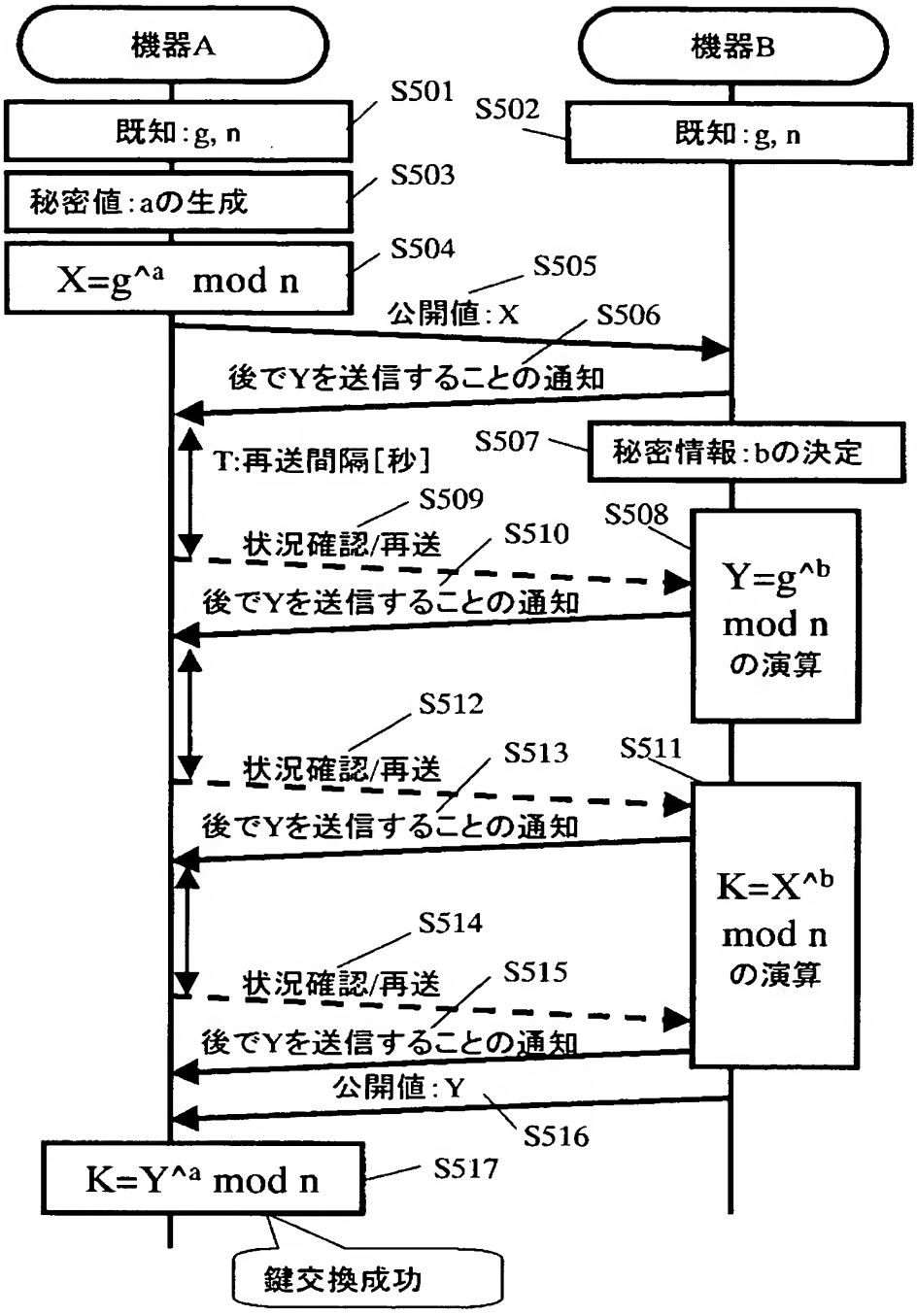
【図３】


端末B-1,
B-2
307
LAN
I/F
305
通信
プロトコル
処理部
303
DB部
304
暗号化/認証処理部
暗号処理
復号処理
301
共有鍵交換部
302
DB管理部
202
GW型機器B
306
通信
プロトコル
処理部
WAN
I/F
308
公衆網
経由
GW型
機器A
鍵交換用の通信
平文(通常データ)の通信
暗号化済みデータの通信

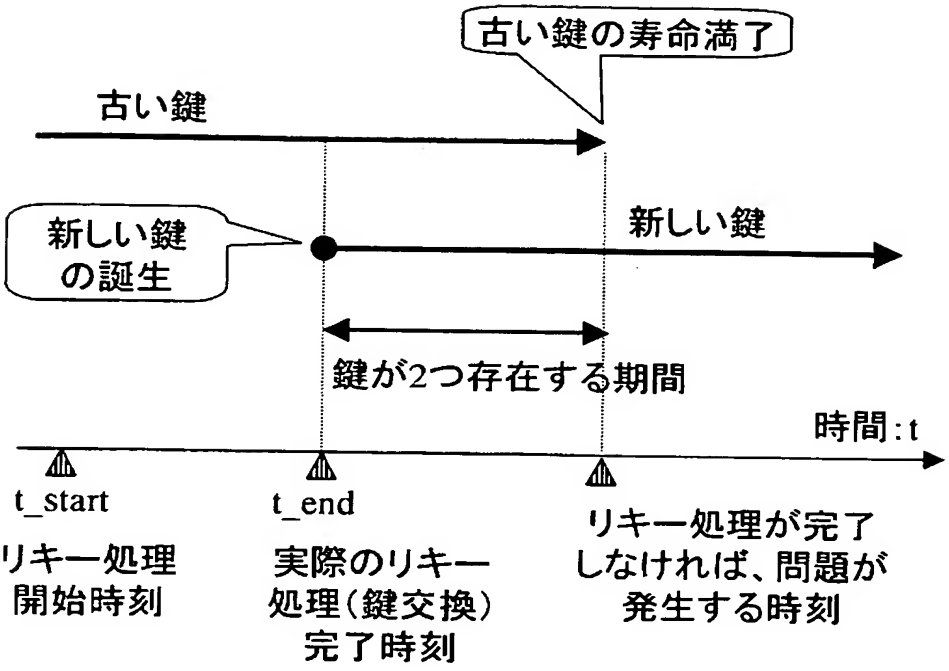
【図４】


機器A
機器B
S401
所要時間の問い合わせ要求
S402
YとKの演算所要時間=Tbを推定
S403
Tb
S405
既知：g, n
S404
既知：g, n
S406
秘密値：aの生成
S407
X=g^a mod n
S408
公開値：X
S409
秘密値：bの生成
S412
S410
Y=g^b mod n の演算
Tb未満
最大Tb時間まで待つ
S411
K=X^b mod n の演算
S413
公開値：Y
S414
K=Y^a mod n
鍵交換成功

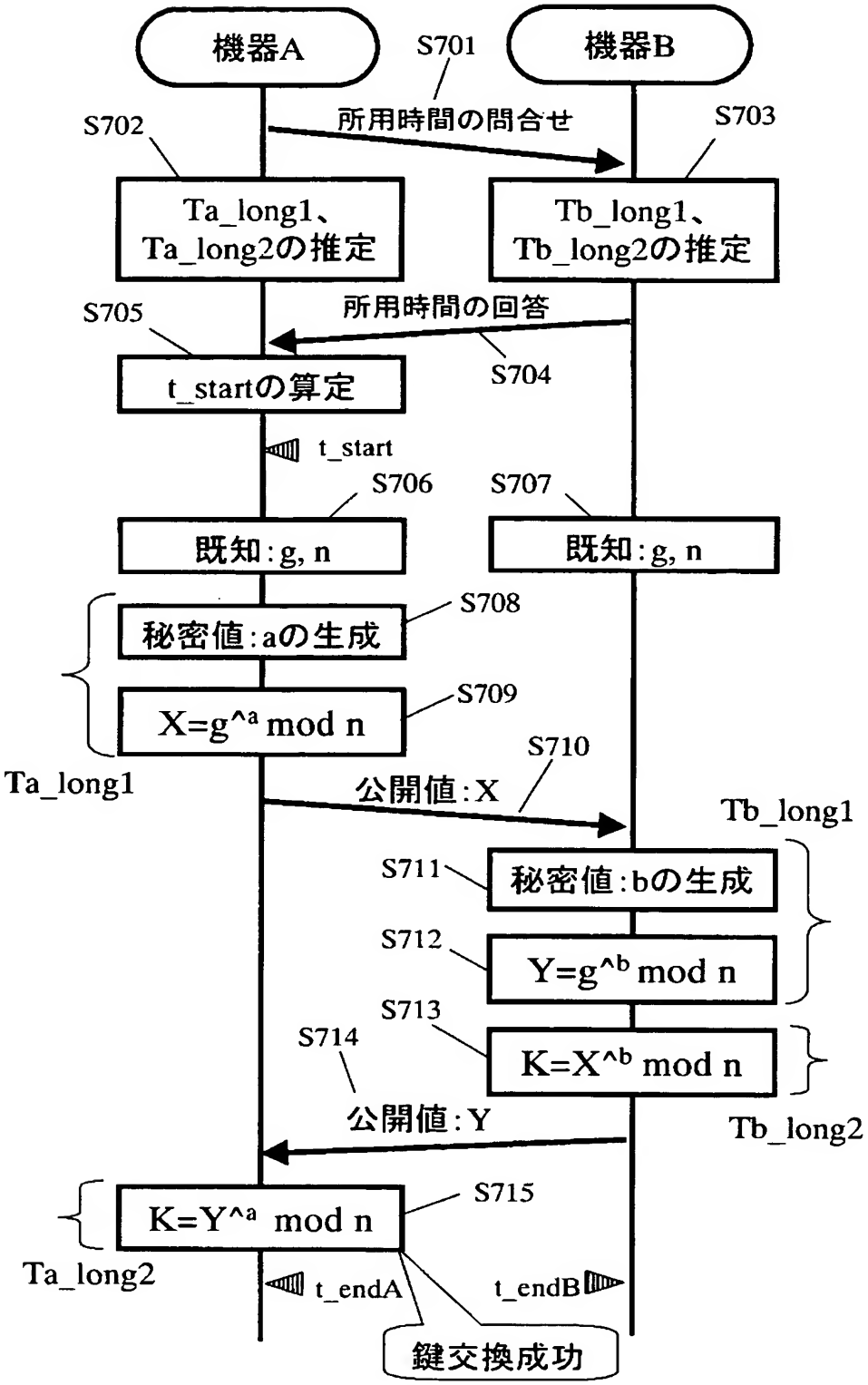
【図５】


機器A
機器B
S501
既知：g, n
S502
既知：g, n
S503
秘密値：aの生成
S504
X=g^a mod n
S505
公開値：X
S506
後でYを送信することの通知
T:再送間隔[秒]
S507
秘密情報：bの決定
S509
状況確認/再送
S510
後でYを送信することの通知
S508
Y=g^b mod n の演算
S512
状況確認/再送
S513
後でYを送信することの通知
S511
K=X^b mod n の演算
S514
状況確認/再送
S515
後でYを送信することの通知
公開値：Y
S516
K=Y^a mod n
S517
鍵交換成功

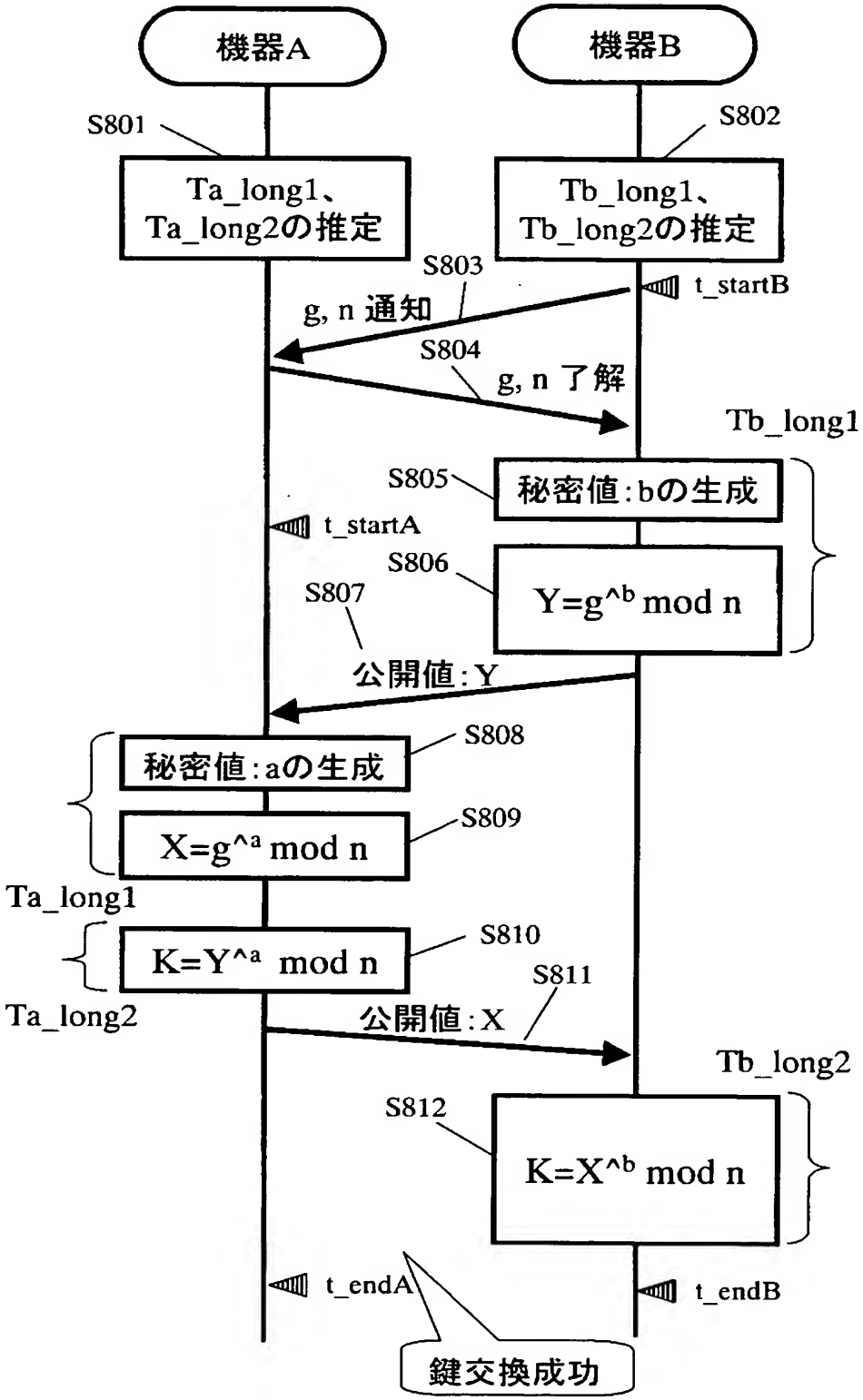
【図６】


古い鍵の寿命満了
古い鍵
新しい鍵
新しい鍵の誕生
鍵が2つ存在する期間
時間:t
t_start
t_end
リキー処理開始時刻
実際のリキー処理(鍵交換)完了時刻
リキー処理が完了しなければ、問題が発生する時刻

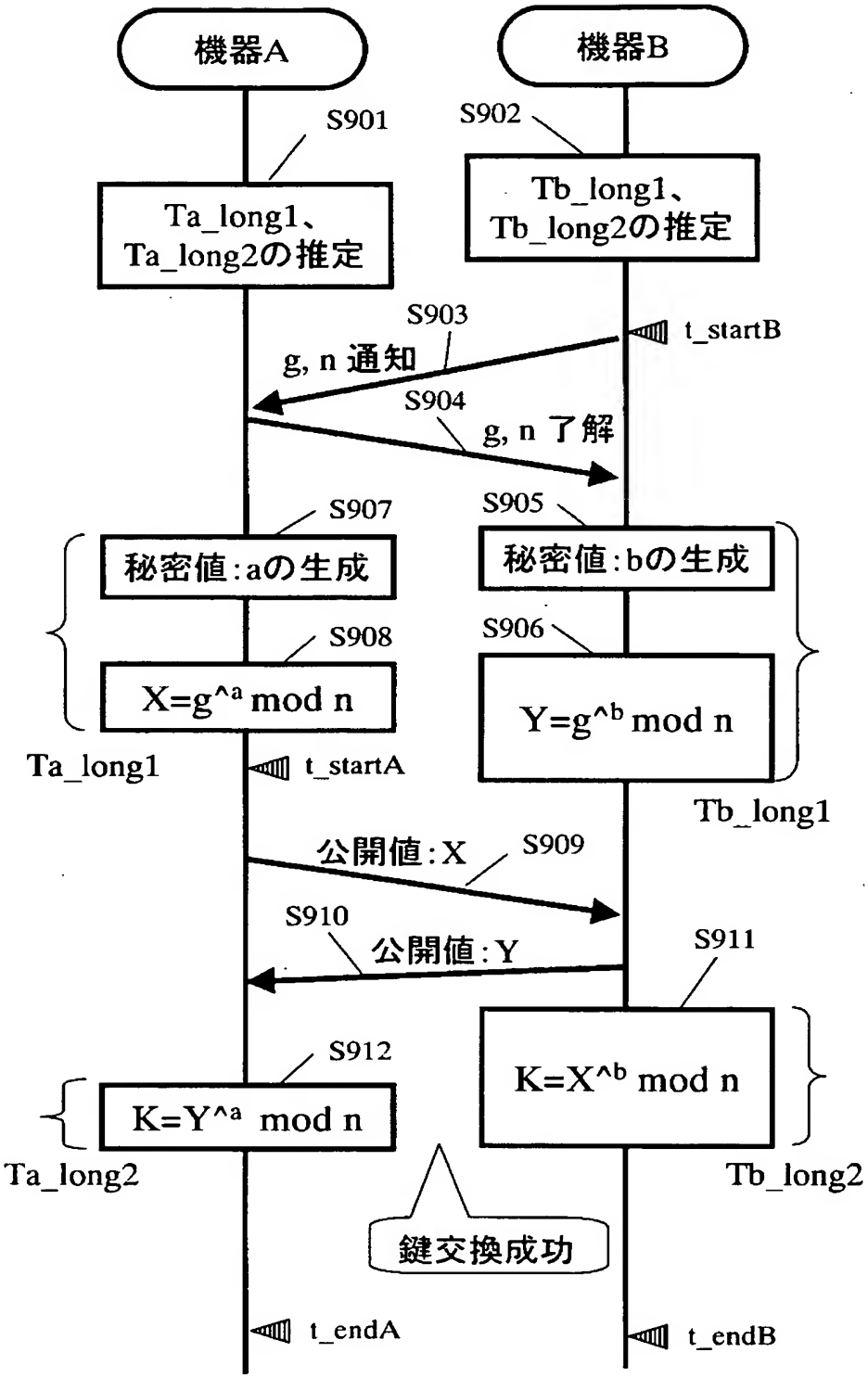
【図７】


機器A
機器B
S701
所用時間の問合せ
S702
S703
Ta_long1、
Ta_long2の推定
Tb_long1、
Tb_long2の推定
S705
所用時間の回答
S704
t_startの算定
t_start
S706
S707
既知：g, n
既知：g, n
S708
秘密値：aの生成
S709
X=g^a mod n
Ta_long1
S710
公開値：X
Tb_long1
S711
秘密値：bの生成
S712
Y=g^b mod n
S713
S714
K=X^b mod n
公開値：Y
Tb_long2
S715
K=Y^a mod n
Ta_long2
t_endA
t_endB
鍵交換成功

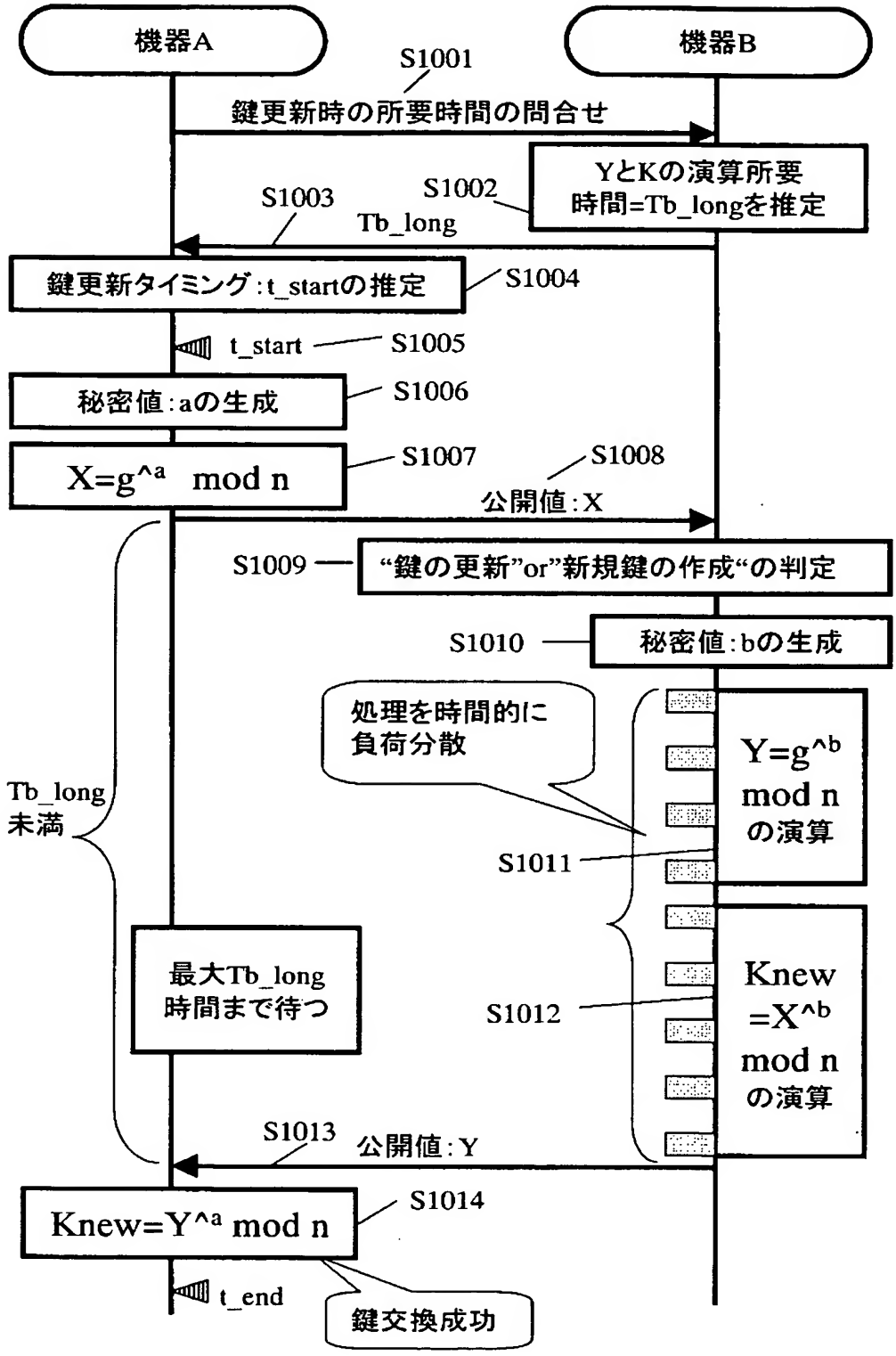
【図８】


機器A
機器B
S801
Ta_long1、
Ta_long2の推定
S802
Tb_long1、
Tb_long2の推定
S803
g, n 通知
t_startB
S804
g, n 了解
Tb_long1
S805
秘密値：bの生成
t_startA
S806
Y=g^b mod n
S807
公開値：Y
S808
秘密値：aの生成
S809
X=g^a mod n
Ta_long1
S810
K=Y^a mod n
S811
Ta_long2
公開値：X
Tb_long2
S812
K=X^b mod n
t_endA
t_endB
鍵交換成功

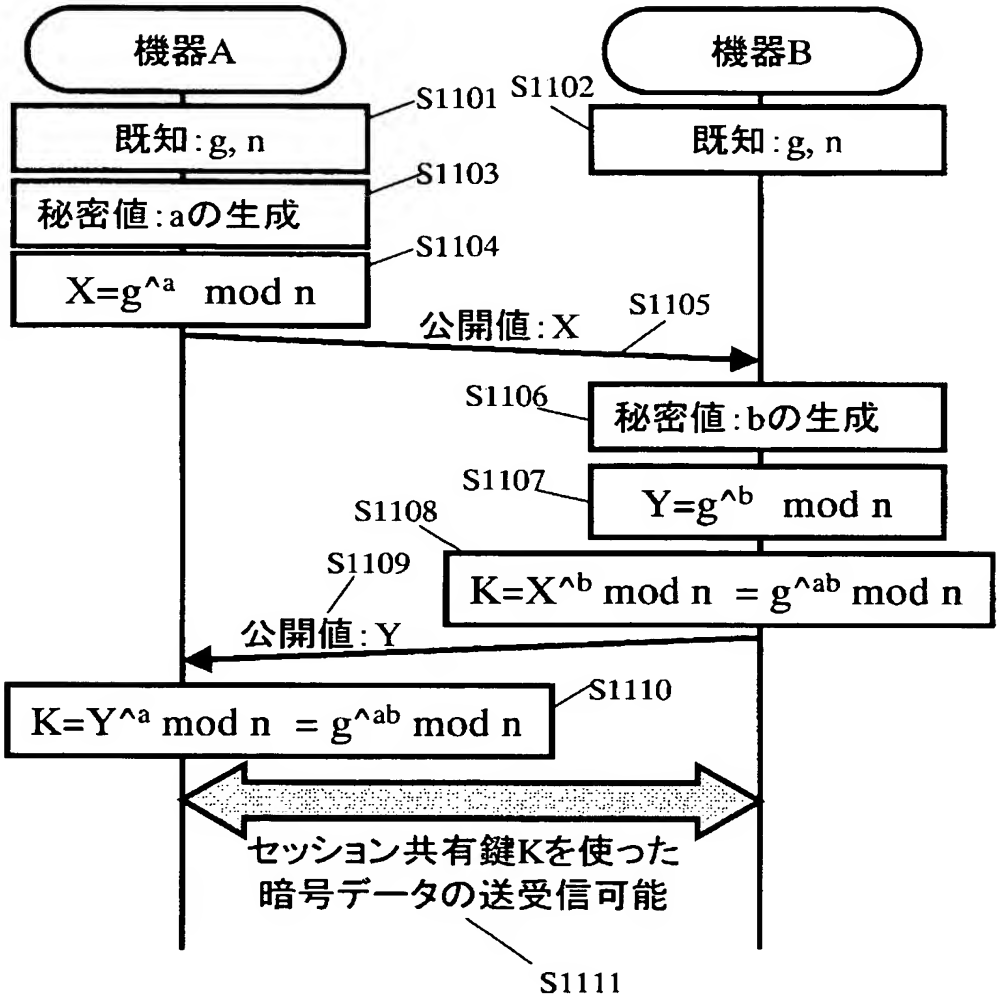
【図９】


機器A
機器B
S901
S902
Ta_long1、
Ta_long2の推定
Tb_long1、
Tb_long2の推定
S903
t_startB
g, n 通知
S904
g, n 了解
S907
S905
秘密値:aの生成
秘密値:bの生成
S908
S906
X=g^a mod n
Y=g^b mod n
Ta_long1
t_startA
Tb_long1
公開値:X
S909
S910
公開値:Y
S911
S912
K=Y^a mod n
K=X^b mod n
Ta_long2
Tb_long2
鍵交換成功
t_endA
t_endB

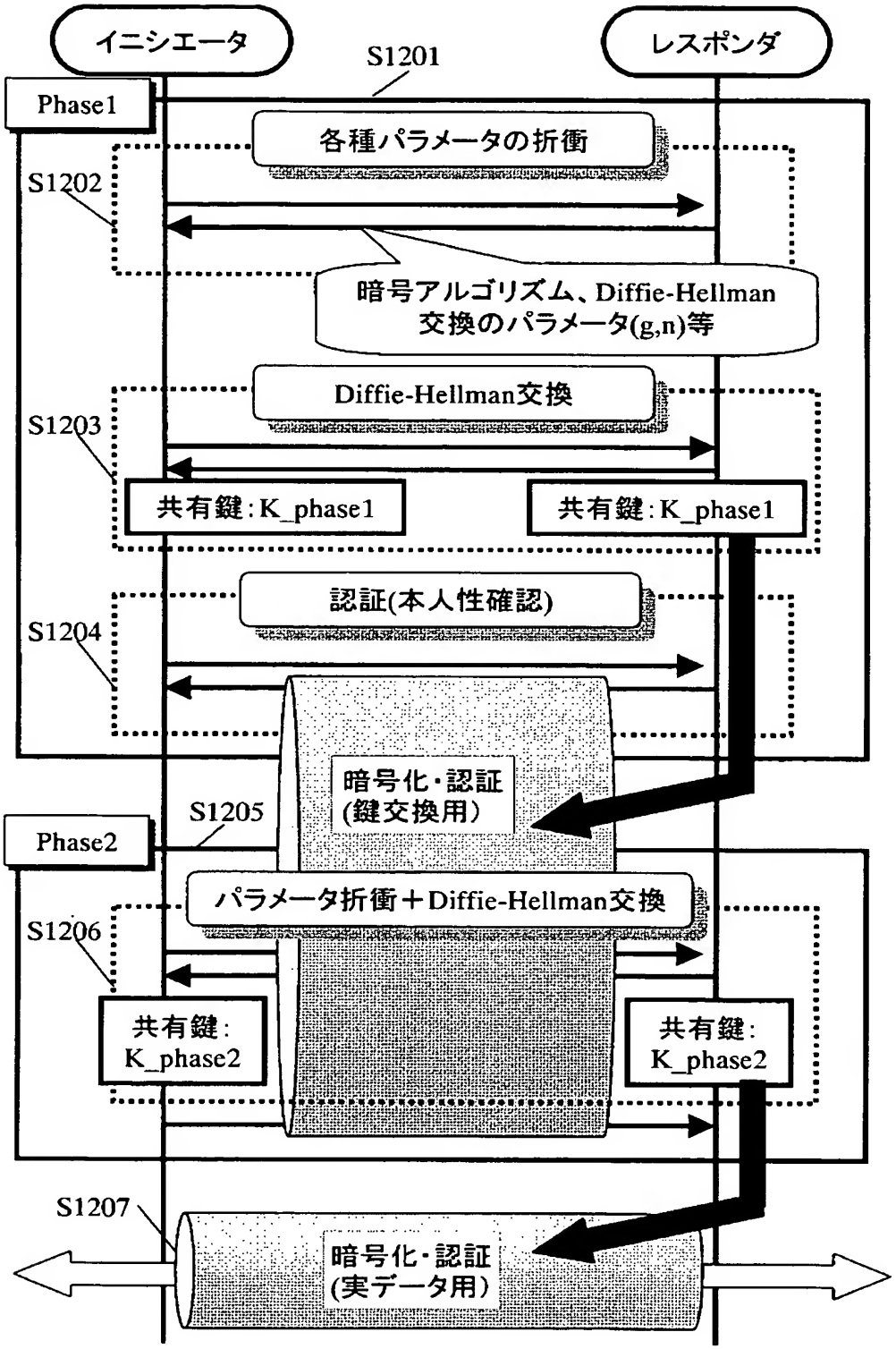
【図１０】


機器A
機器B
S1001
鍵更新時の所要時間の問合せ
YとKの演算所要
時間=Tb_longを推定
S1002
S1003
Tb_long
鍵更新タイミング:t_startの推定
S1004
t_start
S1005
秘密値:aの生成
S1006
X=g^a mod n
S1007
S1008
公開値:X
S1009
“鍵の更新”or”新規鍵の作成“の判定
S1010
秘密値:bの生成
処理を時間的に
負荷分散
Y=g^b
mod n
の演算
S1011
Tb_long
未満
最大Tb_long
時間まで待つ
S1012
Knew
=X^b
mod n
の演算
S1013
公開値:Y
Knew=Y^a mod n
S1014
t_end
鍵交換成功

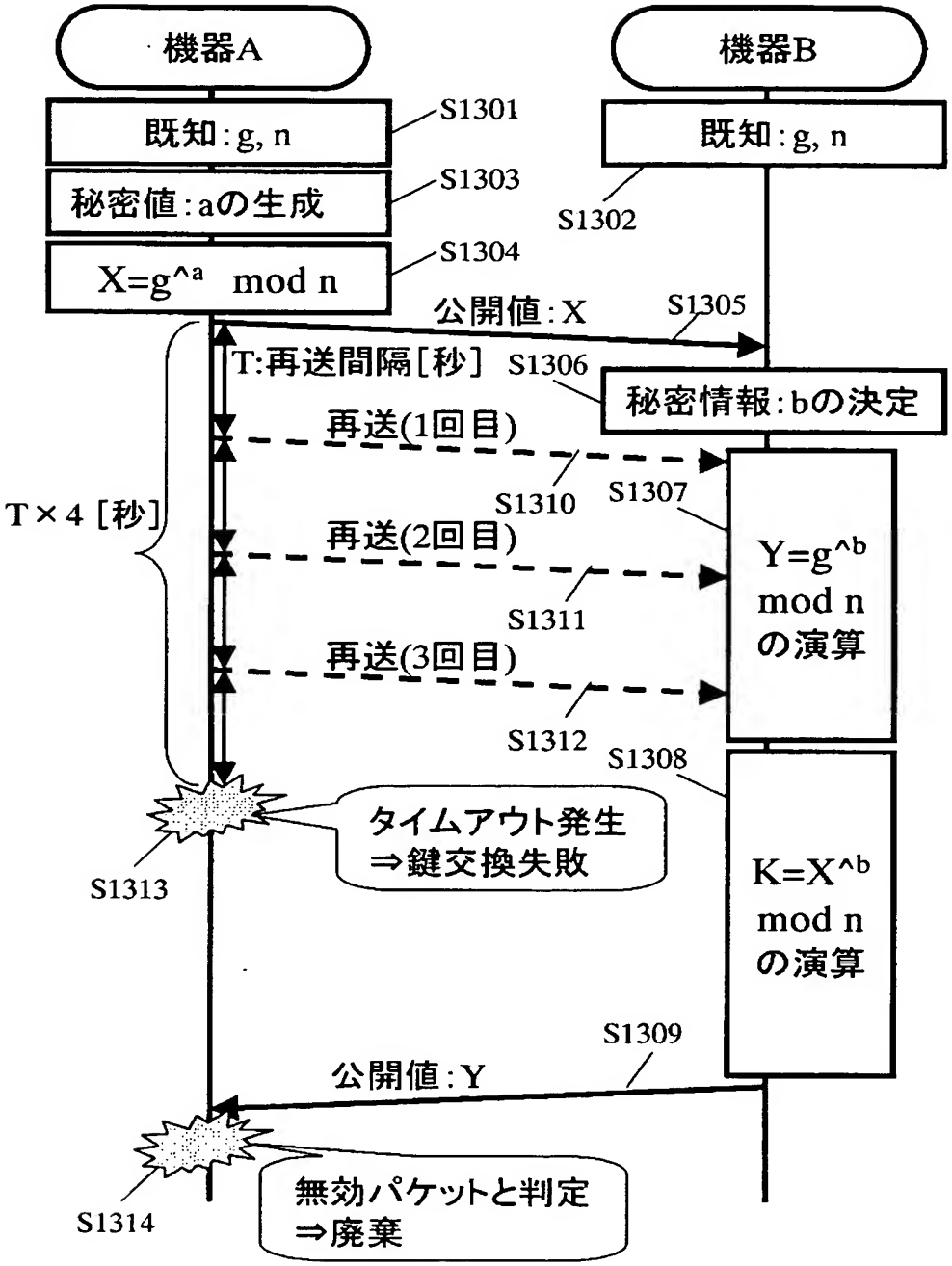
【図１１】


機器A
機器B
S1101
S1102
既知：g, n
既知：g, n
S1103
秘密値：aの生成
S1104
X=g^a mod n
公開値：X
S1105
S1106
秘密値：bの生成
S1107
Y=g^b mod n
S1108
K=X^b mod n = g^ab mod n
S1109
公開値：Y
K=Y^a mod n = g^ab mod n
S1110
セッション共有鍵Kを使った
暗号データの送受信可能
S1111

【図１２】


イニシエータ
レスポンダ
S1201
Phase1
各種パラメータの折衝
S1202
暗号アルゴリズム、Diffie-Hellman
交換のパラメータ(g,n)等
Diffie-Hellman交換
S1203
共有鍵：K_phase1
共有鍵：K_phase1
認証(本人性確認)
S1204
暗号化・認証
(鍵交換用)
S1205
Phase2
パラメータ折衝＋Diffie-Hellman交換
S1206
共有鍵：
K_phase2
共有鍵：
K_phase2
S1207
暗号化・認証
(実データ用)

【図１３】


機器A
機器B
既知:g, n
S1301
既知:g, n
S1303
秘密値:aの生成
S1302
S1304
X=g^a mod n
公開値:X
S1305
T:再送間隔[秒]
S1306
秘密情報:bの決定
再送(1回目)
S1310
S1307
T×4 [秒]
再送(2回目)
Y=g^b mod n の演算
S1311
再送(3回目)
S1312
S1308
タイムアウト発生
⇒鍵交換失敗
S1313
K=X^b mod n の演算
S1309
公開値:Y
無効パケットと判定
⇒廃棄
S1314

【書類名】　要約書

【要約】

【課題】　２つの通信機器の間で暗号化・認証処理されたデータを送受信するシステムにおいて、低性能のＣＰＵを搭載する機器で、鍵交換に伴う高負荷の演算を行う場合にも、対向の機器側でタイムアウトを検出させることなく鍵交換を成功させるような共有鍵交換方法を提供する。

【解決手段】　低性能のＣＰＵを搭載した機器Ｂは、機器Ａから公開値Ｘを受信する（Ｓ１０５）と、自分が相手に応答を返すことのできる応答遅延時間：Ｔｂを推定し、機器Ａに応答遅延時間を予告する（Ｓ１０７）。その後、機器Ｂは鍵交換のための演算を行い、公開値Ｙを算出後機器Ａに送信する（Ｓ１１２）。

【選択図】　図１

特願２００３－０１５８６６

出　願　人　履　歴　情　報

識別番号　　［０００００５８２１］

１．変更年月日　　１９９０年　８月２８日
［変更理由］　　新規登録
住　所　　大阪府門真市大字門真１００６番地
氏　名　　松下電器産業株式会社